

Network Working Group  
Request For Comments: 826

David C. Plummer  
(DCP@MIT-MC)  
Noviembre 1982

## Un Protocolo Para la Resolución de Dirección Ethernet

-- o --

Conversión de Direcciones de Protocolo de Red  
a Dirección Ethernet de 48 bits  
para la Transmisión sobre  
Hardware Ethernet

### Resumen

La implementación de un protocolo P en un host emisor S decide, a través de un mecanismo de enrutamiento del protocolo P, que quiere transmitir a un host de destino T localizado en algún lugar de un cable Ethernet 10Mbit. Para transmitir el paquete Ethernet se debe generar una dirección Ethernet de 48 bits. Las direcciones de hosts dentro de un protocolo P no son siempre compatibles con la correspondiente dirección Ethernet (siendo de diferentes longitudes o valores). Lo presentado aquí es un protocolo que permite la distribución dinámica de la información necesaria para construir tablas para traducir una dirección A en un espacio de direcciones de un protocolo P a direcciones Ethernet de 48 bits.

Se han hecho generalizaciones que permiten al protocolo ser usado por hardware que no sea Ethernet 10Mbit. Algunas redes que envían paquetes por radio son ejemplos de ese hardware.

-----  
El protocolo propuesto aquí es el resultado de un gran acuerdo a partir de un debate con varias personas, los más notables J. Noel Chiapa, Yogen Dalal y James E. Kulp, y comentarios de ayuda de David Moon.

[El propósito de este RFC es presentar un método de Conversión de Direcciones de Protocolo (ej.: direcciones IP) a Direcciones de Red Local (ej.: direcciones Ethernet). Este es un tema de interés general en la comunidad ARPA Internet en este momento. El método propuesto aquí es presentado para su consideración y comentario. Esta no es la especificación de un estándar de Internet.]

### Notas:

-----

Este protocolo fue originalmente diseñado para la Ethernet 10Mbit de DEC/Intel/Xerox. Ha sido generalizado para permitir su uso en otros tipos de redes. Muchas de las discusiones serán directamente sobre la Ethernet 10Mbit. Las generalizaciones, donde sean aplicables, seguirán la discusión específica de Ethernet.

Nos referiremos al protocolo de Internet DOD como Internet.

Los números están en el estándar Ethernet, donde está primero el byte mayor. Esto es opuesto al direccionamiento de byte de máquinas como PDP-11s y VAXes. Por esto, se debe tomar un cuidado especial con el campo opcode ['Operation CODE', código de operación] (ar\$op) descrito a continuación.

Se necesita una autoridad de aceptación superior para manejar los valores del espacio de nombres del hardware (vea continuación). Hasta que exista una autoridad oficial, las solicitudes deben ser enviadas a

David C. Plummer  
Symbolics, Inc.  
243 Vassar Street  
Cambridge, Massachusetts 02139

Alternativamente, los correos electrónicos se pueden enviar a DCP@MIT-MC.

El problema:

-----

El mundo es una jungla en general, y al juego de trabajo en red contribuyen muchos animales. En casi toda capa de una arquitectura de red hay varios protocolos que potencialmente pueden ser usados. Por ejemplo, en un nivel alto, están TELNET y SUPDUP para ingreso remoto. En alguna parte bajo eso está un protocolo confiable de corriente de bytes, que puede ser el protocolo CHAOS, DOD TCP, XEROX BSP o DECnet. Incluso más cerca del hardware está la capa de transporte lógico, que puede ser CHAOS, DOD Internet, Xerox PUP, o DECnet. La Ethernet 10Mbit permite a todos estos protocolos (y más) coexistir en un solo cable por el significado de un campo tipo en el encabezado del paquete Ethernet. Sin embargo, la Ethernet 10Mbit requiere direcciones de 48 bits en un cable físico, aunque la mayoría de las direcciones de protocolo no son de 48 bits de longitud, ni tampoco tienen necesariamente alguna relación con la dirección Ethernet de 48 bits del hardware. Por ejemplo, las direcciones CHAOS son de 16 bits, las direcciones de DOD Internet son de 32 bits, y las direcciones Xerox PUP son de 8 bits. Un protocolo es necesario para distribuir dinámicamente las correspondencias entre un par <protocolo, dirección> y una dirección Ethernet de 48 bits.

Motivación:

-----

El uso de la Ethernet 10Mbit está incrementándose como muchos fabricantes de interfaces conforme a la especificación publicada por DEC, Intel y Xerox. Con esta disponibilidad incrementándose, más y más software está siendo escrito para estas interfaces. Hay dos alternativas: (1) Todo aquel que la implementa inventa su propio método para hacer de alguna forma la resolución de dirección, o (2) todo aquel que la implementa usa un estándar para que su código pueda ser distribuido a otros sistemas sin necesidad de modificación. Esta propuesta intenta establecer el estándar.

Definiciones:

-----

Se define lo siguiente para referirse a los valores puestos en el campo TIPO del encabezado del paquete Ethernet:

ether\_type\$XEROX\_PUP,  
ether\_type\$DOD\_INTERNET,  
ether\_type\$CHAOS,

y uno nuevo:

ether\_type\$ADDRESS\_RESOLUTION.

También se definen los siguientes valores (para ser discutidos luego):  
     ares\_op\$REQUEST (= 1, el byte mayor transmitido primero) y  
     ares\_op\$REPLY (= 2),  
 y  
     ares\_hrd\$Ethernet (= 1).

Formato del paquete:  
 -----

Para comunicar mapeos desde los pares <protocolo, dirección> a direcciones Ethernet de 48 bits, un formato de paquete que contiene el protocolo de Resolución de Dirección es necesario. El formato del paquete se muestra a continuación.

Capa de transmisión Ethernet (no necesariamente accesible para el usuario):

48 bits: Dirección Ethernet de destino  
 48 bits: Dirección Ethernet del emisor  
 16 bits: Tipo de protocolo = ether\_type\$ADDRESS\_RESOLUTION

Datos del paquete Ethernet:

16 bits: (ar\$hrd) Espacio de dirección de hardware (ej: Ethernet, red de paquetes de difusión).  
 16 bits: (ar\$pro) Espacio de dirección del protocolo. Para el hardware Ethernet, esta está en la configuración de campos tipo

ether\_type\$<protocolo>.

8 bits: (ar\$hln) Largo en bytes de cada dirección de hardware.  
 8 bits: (ar\$pln) Largo en bytes de cada dirección de protocolo.  
 16 bits: (ar\$op) opcode (ares\_op\$REQUEST | ares\_op\$REPLY).  
 nbytes: (ar\$sha) Dirección de hardware del emisor de este paquete, n es tomada del campo ar\$hln.  
 mbytes: (ar\$spa) Dirección de protocolo del emisor de este paquete, m es tomada del campo ar\$pln.  
 nbytes: (ar\$tha) Dirección de hardware del destinatario de este paquete (si la sabe).  
 mbytes: (ar\$tpa) Dirección de protocolo del destinatario.

Generación de paquete:  
 -----

Como un paquete es enviado a través de las capas de red, el encaminamiento determina la dirección de protocolo del próximo salto para el paquete y en que pieza de hardware espera encontrar la estación con la dirección de protocolo de destino. En el caso de la Ethernet 10Mbit, la resolución de dirección es necesaria y alguna capa menor (probablemente el controlador de hardware) debe consultar al módulo de Resolución de Dirección (quizás implementado en el módulo de soporte Ethernet) para convertir el par <tipo de protocolo, dirección protocolo de destino> a la dirección Ethernet de 48 bits. El módulo de Resolución de Dirección intenta encontrar este par en una tabla. Si encuentra el par, devuelve la correspondiente dirección Ethernet de 48 bits al solicitante (el controlador de hardware) que entonces transmite el paquete. Si no, probablemente informa al solicitante que está desechando el paquete (se supone que el paquete será retransmitido por una capa de red superior), y genera un paquete Ethernet con un campo tipo de ether\_type\$ADDRESS\_RESOLUTION. El módulo de Resolución de Dirección entonces establece el campo ar\$hrd a ares\_hrd\$Ethernet, ar\$pro al tipo de protocolo que está siendo resuelto, ar\$hln a 6 (el número de bytes en una dirección Ethernet 48 bits), ar\$pln a la longitud de una dirección de este protocolo, ar\$op a ares\_op\$REQUEST, ar\$sha con la dirección

Ethernet de 48 bits de sí mismo, `ar$spa` con la dirección de protocolo de sí mismo, y `ar$tpa` con la dirección de protocolo de la máquina que está tratando de ser accedida. No establece `ar$tha` a nada en particular, porque este es el valor que está tratando de determinar. Puede establecer `ar$tha` a la dirección de difusión para el hardware (todos unos en el caso de Ethernet 10Mbit) si esto se hace conveniente para algún aspecto de la implementación. Esto causa entonces que el paquete sea difundido a todas las estaciones en el cable Ethernet originalmente determinado por el mecanismo de encaminamiento.

#### Recepción de paquete:

-----

Cuando un paquete de resolución de dirección es recibido, el módulo Ethernet receptor envía el paquete al módulo de Resolución de Dirección que lo pasa a través de un algoritmo similar al siguiente. Las respuestas negativas indican un fin del proceso y un descarte del paquete.

¿Tengo el tipo de hardware en `ar$hrd`?

Si: (casi definitivamente)

[opcionalmente chequea el largo de hardware `ar$hln`]

¿Hablo el protocolo en `ar$pro`?

Si:

[opcionalmente chequea el largo de protocolo `ar$pln`]

`Merge_flag` := falso

Si el par <tipo protocolo, dirección de protocolo del emisor> ya está en mi tabla de traducción, actualizo el campo de dirección de hardware del emisor de la entrada con la nueva información en el paquete y pongo `Merge_flag` en verdadero.

¿Soy la dirección protocolo de destino?

Si:

Si `Merge_flag` es falso, agrego el trío <tipo protocolo, dirección protocolo del emisor, dirección hardware del emisor> a la tabla de traducción. ¿Es el opcode `ares_op$REQUEST`? (¡¡AHORA ve en el opcode!!)

Si:

Cambio los campos hardware y protocolo, poniendo las direcciones hardware y protocolo local en el campo del emisor.

Configuro el campo `ar$op` a `ares_op$REPLY`

Envío el paquete a la (nueva) dirección hardware de destino por el mismo hardware por la que solicitud fue recibida.

Observe que el trío <tipo protocolo, dirección protocolo del emisor, dirección hardware del emisor> está incluido dentro de la tabla antes que el opcode sea visto. Asumiendo que la comunicación es bidireccional; si A tiene alguna razón para hablar con B, entonces B probablemente tendrá alguna razón para hablar con A. Observe también que si una entrada ya existe para el par <tipo protocolo, dirección protocolo del emisor>, entonces la nueva dirección de hardware reemplaza a la antigua. Aspectos Relacionados da algunas ideas para esto.

Generalización: Los campos `ar$hrd` y `ar$hln` permiten a este protocolo y al formato del paquete ser usados para otras redes que no sean Ethernets 10Mbit. Para la Ethernet 10Mbit se toma <`ar$hrd`, `ar$hln`> en el valor <1, 6>. Para otros hardware de redes, el campo `ar$pro` puede no

corresponder al campo tipo Ethernet, pero debe ser asociado con el

protocolo cuya resolución de dirección esta siendo realizada.

¿Por qué es hecho de esta manera?

-----

No se desea en ningún caso hacer difusiones periódicas. Imagine 100 estaciones de trabajo en una Ethernet, cada una difundiendo una información de resolución de dirección cada 10 minutos (como un conjunto posible de parámetros). Esto es un paquete cada 6 segundos. Esto es bastante razonable, ¿pero qué uso tiene? Las estaciones de trabajo no están generalmente hablando con cada una de las demás (y entonces tienen 100 entradas sin uso en una tabla); estarán principalmente hablando con un mainframe, un servidor de archivos o un bridge, pero sólo con un pequeño número de otras estaciones de trabajo (para conversaciones interactivas, por ejemplo). El protocolo descrito en este texto distribuye información como es necesario, y una sola vez (probablemente) por arranque de una máquina.

Este formato no permite que se haga más de una resolución en el mismo paquete. Esto es por simplicidad. Si los objetos fueran multiplexados el formato del paquete sería considerablemente más pesado de digerir, y mucha de la información puede ser injustificada. Piense en un bridge que habla cuatro protocolos diciéndole a una estación de trabajo las cuatro direcciones de protocolo, tres de estas la estación de trabajo probablemente nunca use.

Este formato permite al buffer del paquete ser reusado si una respuesta es generada; una respuesta tiene la misma longitud que una solicitud, y varios de los campos son los mismos.

El valor del campo hardware (ar\$hrd) se toma de una lista para este propósito. Actualmente el único valor definido es para las Ethernet 10Mbit (ares\_hrd\$Ethernet = 1). Se ha hablado de usar este protocolo para Redes de Difusión de Paquetes, y esto requerirá otro valor para otro futuro medio de hardware que desee usar este protocolo.

Para Ethernet 10Mbit, el valor en el campo protocolo (ar\$pro) es tomado del conjunto ether\_type\$. Este es un reuso del tipo de protocolo asignado. Si se combinase con el opcode (ar\$op) efectivamente dividiría a la mitad el número de protocolos que pueden ser resueltos bajo este protocolo y haría que un monitor/depurador sea más complejo (vea Monitoreo y Depuración de Red debajo). Esperamos nunca ver 32768 protocolos, pero Murphy hizo algunas leyes que no nos permiten hacer esta suposición.

En teoría, los campos de longitud (ar\$hln y ar\$pln) son redundantes, la longitud de una dirección de protocolo debe estar determinada por el tipo de hardware (encontrado en ar\$hrd) y el tipo de protocolo

(encontrado en ar\$pro). Esto es incluido para un chequeo de consistencia opcional, y para monitoreo y depuración de red (vea debajo).

El opcode está para determinar si es una solicitud (que puede causar una respuesta) o una respuesta a una solicitud previa. Los 16 bits para esto

son excesivos, pero una bandera (campo) es necesaria.

La dirección de hardware del emisor y la dirección de protocolo del emisor son absolutamente necesarias. Estos son los campos que se ponen en una tabla de traducción.

La dirección protocolo de destino es necesaria en la solicitud del paquete para que una máquina puede determinar si ingresa o no la información del emisor en una tabla o para enviar una respuesta. Esto no es esencialmente necesario en la respuesta si uno asume que una respuesta es solo provocada por una solicitud. Esto es incluido para integridad, monitoreo de red, y para simplificar el algoritmo de proceso sugerido descripto anteriormente (que no ve el opcode hasta DESPUÉS de poner la información del emisor en una tabla).

La dirección de hardware de destino es incluida para integridad y para monitoreo de red. Esto no tiene sentido en la solicitud, es por este número que la máquina es solicitada. Su significado en una respuesta es la dirección de la máquina haciendo la solicitud. En algunas implementaciones (que no miran en el encabezado ethernet de 14 bytes, por ejemplo) esto puede salvar algún registro lento o ahorrar espacio en el stack enviando este campo al controlador de hardware como la dirección hardware de destino del paquete.

No hay bytes de relleno entre direcciones. El paquete de información debe ser visto como una trama de bytes en que solo 3 pares de bytes son definidos como palabras (ar\$hrd, ar\$pro y ar\$op) donde son enviados primero los bytes más significativos (estilo de byte Ethernet/PDP-10).

Monitoreo y depuración de red:

-----

El protocolo de Resolución de Dirección anterior permite a una máquina conocer sobre la actividad del protocolo de nivel superior (ej: CHAOS, Internet, PUP, DECnet) en un cable Ethernet. Puede determinar que campos de tipo de protocolo Ethernet están en uso (por valor) y las direcciones de protocolo en cada tipo de protocolo. En realidad, no es necesario para el monitor hablar alguno de los protocolos de nivel superior involucrados. Sería algo así como esto:

Cuando un monitor recibe un paquete de Resolución de Dirección, este siempre ingresa el <tipo protocolo, dirección protocolo del emisor,

dirección hardware del emisor> en una tabla. Puede determinar la longitud de la dirección de hardware y de protocolo desde los campos ar\$hln y ar\$p1n del paquete. Si el opcode es una RESPUESTA el monitor puede entonces desechar el paquete. Si el opcode es una SOLICITUD y la dirección de protocolo de destino coincide con la dirección de protocolo del monitor, el monitor envía una RESPUESTA como normalmente haría. El monitor sólo tomará un mapeo de este manera, luego la RESPUESTA a la SOLICITUD será enviada directamente al host solicitante. El monitor puede intentar enviar sus propias SOLICITUDES, pero puede haber dos monitores dentro de una ciclo de envíos de SOLICITUD, se debe tener cuidado.

Debido a que el protocolo y el opcode no están combinados dentro de un campo, el monitor no necesita saber que solicitud opcode está asociada con que respuesta opcode por el mismo protocolo de nivel superior. Los campos de longitud deben también dar suficiente información para activar el "análisis" de unas direcciones de protocolo, aunque no tienen conocimiento de que significan las direcciones de protocolo.

Una implementación de trabajo del Protocolo de Resolución de Dirección puede también ser usada para depurar una implementación que no trabaje. Probablemente un controlador de hardware difundirá satisfactoriamente un paquete con el campo tipo Ethernet de ether\_type\$ADDRESS\_RESOLUTION. El formato del paquete puede no ser totalmente correcto, porque las implementaciones iniciales pueden tener bugs, y el manejo de la tabla puede ser ligeramente tramposo. Debido a que las solicitudes son difusiones un monitor recibirá el paquete y podrá mostrarlo para depuración si es deseado.

Un ejemplo:

-----

Existen máquinas X e Y que están en el mismo cable Ethernet 10Mbit. Ellas tienen direcciones Ethernet EA(X) y EA(Y) y direcciones Internet DOD IPA(X) e IPA(Y). El tipo Ethernet de Internet es ET(IP). La máquina X ha sido recién iniciada, e inmediatamente o luego querrá enviar un paquete Internet a la máquina Y en el mismo cable. X sabe que necesita enviar a IPA(Y) y llama al controlador de hardware (aquí un controlador Ethernet) IPA(Y). El controlador consulta al módulo de Resolución de Dirección para convertir <ET(IP), IPA(Y)> a una dirección Ethernet de 48 bits, pero como X fue recién iniciada, no tiene esta información. Desecha el paquete Internet y en su lugar crea un paquete de RESOLUCIÓN DE DIRECCIÓN con

```
(ar$hrd) = ares_hrd$Ethernet
(ar$pro) = ET(IP)
(ar$hln) = length (EA(X))
(ar$pln) = length (IPA(X))
(ar$op) = ares_op$REQUEST
```

```
(ar$sha) = EA(X)
(ar$spa) = IPA(X)
(ar$tha) = no importa
(ar$tpa) = IPA(Y)
```

y difunde este paquete a todos en el cable.

La máquina Y toma este paquete y determina que entiende el tipo de hardware (Ethernet), que habla el protocolo indicado (Internet) y que el paquete es para ella ((ar\$tpa)=IPA(Y)). Ingresa (probablemente reemplazando alguna entrada existente) la información que mapea <ET(IP), IPA(X)> a EA(X). Entonces observa que es una solicitud, cambia los campos, poniendo EA(Y) en el nuevo campo dirección Ethernet del emisor (ar\$sha), configura el opcode a respuesta, y envía el paquete directamente (no difundido) a EA(X). En este punto Y sabe como enviar a X, pero X todavía no sabe como enviar a Y.

La máquina X toma el paquete de respuesta de Y, forma el mapeo de <ET(IP), IPA(Y)> a EA(Y), observa que el paquete es una respuesta y lo desecha. La próxima vez que el módulo Internet de X intente enviar un paquete a Y en la Ethernet, la traducción será exitosa, y el paquete (expectante) arribará. Si el módulo de Internet de Y entonces quiere hablar a X, esto también será exitoso desde que Y ha recordado la información de la solicitud de X para la Resolución de Dirección.

Aspectos relacionados:

-----

Puede ser conveniente tener una tabla con temporizadores y/o timeouts. La implementación de esto está fuera del ámbito de este protocolo. Aquí hay una descripción más detallada (gracias a MOON@SCR@MIT-MC).

Si un host se mueve, alguna conexión iniciada por este host trabajará, asumiendo que su propia tabla de resolución de dirección es vaciada cuando se mueve. Sin embargo, las conexiones iniciadas con el por otros hosts no tendrán una razón particular para saber que deben desechar sus viejas direcciones. Sin embargo, las direcciones Ethernet de 48 bits supuestamente son únicas y fijas para siempre, entonces ellas no deben cambiar. Un host se debe "mover" si un nombre de host (y dirección en algún otro protocolo) fue reasignado a una pieza física de hardware diferente. También, como sabemos por experiencia, está siempre el peligro de que se tome información enrutada de manera incorrecta transmitida accidentalmente por un error de hardware o software; no se debe permitir que esta información persista por siempre. Quizás un fallo al iniciar una conexión debe informar al módulo de Resolución de Dirección que el host no es alcanzable para borrar la información en la base, posiblemente porque este no está funcionando o la vieja traducción ya no es válida. O quizás la recepción de un paquete desde un host debe reiniciar un temporizador en la entrada de la resolución de dirección

usada para transmitir paquetes a este host; si no son recibidos paquetes desde un host por un período conveniente de tiempo, la entrada de resolución de dirección es olvidada. Esto puede causar gasto extra para revisar la tabla por cada paquete entrante. Quizás un hash o un índice puede hacer esto más veloz.

El algoritmo sugerido para la recepción de paquetes de resolución de dirección intenta disminuir el tiempo que toma recuperar si un host se mueve. Si el <tipo de protocolo, dirección de protocolo del emisor> ya está en la tabla de traducción, entonces la dirección de hardware del emisor sobreescribe la entrada existente. De esta forma, en una Ethernet perfecta donde una SOLICITUD difundida alcanza todas las estaciones en el cable, cada estación tomará la nueva dirección de hardware.

Otra alternativa es tener un demonio realizando los timeouts. Después de un tiempo conveniente, el demonio considera la remoción de una entrada. Primero envía (con un número pequeño de retransmisiones si es necesario) un paquete de resolución de dirección con el opcode de SOLICITUD directamente a la dirección Ethernet en la tabla. Si no es vista una RESPUESTA en un período corto de tiempo la entrada es borrada. La solicitud es enviada directamente, no como una molestia, para cada estación en la Ethernet. Olvidar las entradas directamente puede causar a lo mejor que información útil, que luego deberá ser recobrada, sea olvidada.

Desde los hosts sólo se transmite información sobre ellos mismos, el reinicio de un host causará que su tabla de mapeo de direcciones esté al día. La mala información no puede persistir por siempre siendo pasada de máquina en máquina; la única mala información que puede existir está en una máquina que no sabe que alguna otra ha cambiado su dirección Ethernet de 48 bits. Quizás el reinicio (o limpieza) manual de la tabla de mapeo de dirección alcanzará.

Este aspecto claramente necesita más opinión si es considerado importante. Esto es motivado por algún protocolo como el de resolución de dirección.

Traducción al castellano: Javier Waisbrot (2002)