**Linux Networking:**

Address Resolution Protocol

David Morgan

---

"Hardware address" to "Protocol address" translation

- Network layer and up use one addressing scheme
- Data link and down use (if any) another
- Network-up: “protocol” addresses
- Datalink-down: “hardware” addresses
“Hardware” vs “Protocol” addresses

- **Protocol addresses**
  - software abstractions
  - apps use them to identify destination computers
  - hardware cannot locate a computer using one

- **Hardware addresses**
  - applications don’t use them
  - hardware can locate a computer using one
  - but only within same physical net (computers on common medium)

---

**Example**

- **IP addresses**
  - 32-bit numbers
  - telnet/ftp/http use them to identify destination computers
  - ethernet cannot locate a computer using one

- **Ethernet addresses**
  - 48-bit numbers
  - telnet/ftp/http don’t use them
  - ethernet can locate a computer on the common coax or hub using one
Translation necessary

- Given an IP destination, what is the matching ethernet address?
- Address Resolution Protocol finds out (resolves)

---

Ethernet frame structure

<table>
<thead>
<tr>
<th>Source HWaddr</th>
<th>Destination HWaddr</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ethernet’s Data Payload</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Packet Checksum</td>
</tr>
</tbody>
</table>

© David Morgan 2003-14
Ethernet NICs’ reading habits
-- frames that NICs read

- Frames with the NIC’s own address
- Frames with the address FF:FF:FF:FF:FF:FF
- Others ignored (payload never read)

Quick quiz

1. What address gets a frame read by all receiving NICs?
2. What is that address called?
Answers to quiz

1. FF:FF:FF:FF:FF:FF
2. the broadcast address

Ethernet broadcast

<table>
<thead>
<tr>
<th>Source HWaddress</th>
<th>FF:FF:FF:FF:FF:FF</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet’s Data Payload</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packet Checksum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How could we translate?

- Table lookup
  - bindings/mappings kept in memory table
- Message exchange
  - dynamic message exchange across network
- ARP uses both

A lookup table

<table>
<thead>
<tr>
<th>IP address</th>
<th>Ethernet address</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.3.1</td>
<td>00:80:C8:E2:AF:61</td>
</tr>
<tr>
<td>192.168.3.2</td>
<td>00:A0:CC:D2:F0:42</td>
</tr>
<tr>
<td>192.168.3.3</td>
<td>00:40:05:A3:42:26</td>
</tr>
<tr>
<td>192.168.3.4</td>
<td>0A:07:4B:12:82:36</td>
</tr>
<tr>
<td>192.168.3.5</td>
<td>0A:77:81:0E:52:FA</td>
</tr>
</tbody>
</table>
… or how about message exchange?

**Ethernet carrying ARP**

<table>
<thead>
<tr>
<th>Source HWAddress</th>
<th>Destination HWAddress</th>
<th>0806</th>
</tr>
</thead>
</table>

ARP message

Packet Checksum

Ethernet’s payload may be an Address Resolution Protocol message

---

**ARP message structure**

<table>
<thead>
<tr>
<th>HW address type</th>
<th>Protocol address type</th>
</tr>
</thead>
<tbody>
<tr>
<td>HALen</td>
<td>PALen</td>
</tr>
<tr>
<td>Sender HAddr</td>
<td>Target HAddr</td>
</tr>
<tr>
<td>Sender PAddr</td>
<td>Target PAddr</td>
</tr>
</tbody>
</table>

4 bytes

© David Morgan 2003-14
### Ethernet carrying ARP

<table>
<thead>
<tr>
<th>Source HWaddr</th>
<th>Destination HWAddr</th>
<th>0806</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HW address type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protocol address type</td>
<td></td>
</tr>
<tr>
<td>HALen</td>
<td>PALen</td>
<td>Operation</td>
</tr>
<tr>
<td>Sender HAddr</td>
<td>Sender PAddr</td>
<td></td>
</tr>
<tr>
<td>Sender PAddr (cont)</td>
<td>Target HAddr</td>
<td></td>
</tr>
<tr>
<td>Target PAddr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packet Checksum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© David Morgan 2003-14

### B arps (seeks) D

© David Morgan 2003-14
B’s arp request is broadcast...

…reaches everybody; everybody reads it, nobody ignores it

D’s arp reply is direct to B...

…reaches everybody; B reads it, everybody else ignores it
What does “ignore” mean?
What does “read” mean?
who reads and ignores?

- ethernet and arp – separate software entities
- they operate independently
- B’s ethernet may ignore (discard) or accept (to arp)
- B’s arp may then ignore (discard) or reply
Disposition of communiqués

B’s request:

<table>
<thead>
<tr>
<th></th>
<th>at A</th>
<th>at C</th>
<th>at D</th>
<th>at E</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth</td>
<td>reads</td>
<td>reads</td>
<td>reads</td>
<td>reads</td>
</tr>
<tr>
<td>arp</td>
<td>ignores</td>
<td>ignores</td>
<td>reads</td>
<td>ignores</td>
</tr>
</tbody>
</table>

D’s reply:

with **hub**

<table>
<thead>
<tr>
<th></th>
<th>at A</th>
<th>at B</th>
<th>at C</th>
<th>at E</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth</td>
<td>ignores</td>
<td>reads</td>
<td>ignores</td>
<td>ignores</td>
</tr>
<tr>
<td>arp</td>
<td>n/a</td>
<td>reads</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

with **switch**

<table>
<thead>
<tr>
<th></th>
<th>at A</th>
<th>at B</th>
<th>at C</th>
<th>at E</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth</td>
<td>n/a</td>
<td>reads</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>arp</td>
<td>n/a</td>
<td>reads</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Caching arp responses

- arp is inefficient
- takes 3 frames to transfer 1 packet
- packets between host pairs occur in bunches
- so arp caches a table of recent arp’d bindings in memory
- subsequent packets use table, not message exchange
Cached arp table

```
[root@EMACH1 david]# arp -n
Address   HWtype HWaddress    Flags Mask Iface
192.168.3.1 ether 00:80:C8:E2:AF:61 C           eth0
192.168.3.3 ether 00:40:05:A3:42:26 C           eth0
64.130.228.62 ether 00:10:E8:09:6E:80 C           eth1
```

Operation essentials: arp request

- target receives, reads broadcast frame
- caches sender’s addr binding
- compares target IP with his own
  - quit if no match, otherwise...
- compose arp response
  - reverse sender, target addr bindings
  - insert ethernet addr into Sender Haddr field
  - insert “2” (response) in operation field
  - send
Operation essentials: arp reply

- target receives, reads unicast frame
- caches sender’s addr binding
- uses its hardware address to frame and send protocol packet to sender (remember, arp reply “sender” is protocol’s intended “recipient”)