Binding Protocol Addresses (ARP)

BY DR NTALASHA

Content

Content - 2

- **ARP Message Delivery**
- **ARP Message Protocol**
- **Sending an ARP Message**
- **Identifying ARP Frames**
- **Caching ARP Responses**
- **Processing An Incoming ARP Message**
- **Layering, Address Resolution, Protocol Addresses**

Introduction

- To transmit a packet across a LAN or WAN link, a host or router needs to have the hardware address of the intended recipient.
- When an Internet router receives an IP packet, it looks up a next-hop *IP address* in its routing table.
- Routing tables do not tell a router what *hardware* address to use for the next hop.
- Hosts and routers need to have a method of translating an IP address into a hardware address.
- This unit is about various ways to do that translation.
- The translation is called *address resolution*

Protocol Addresses And Packet Delivery

- The Internet uses certain standard kinds of addresses and packet formats.
- The actual physical networks over which the Internet traffic flows use different addressing schemes and frame formats -- incompatible with TCP/IP standards and generally with the other physical networks too.
- When Internet packets are sent over network links, they are *encapsulated* - they travel *inside* frames appropriate to the physical network, addressed using the addressing scheme of that network.

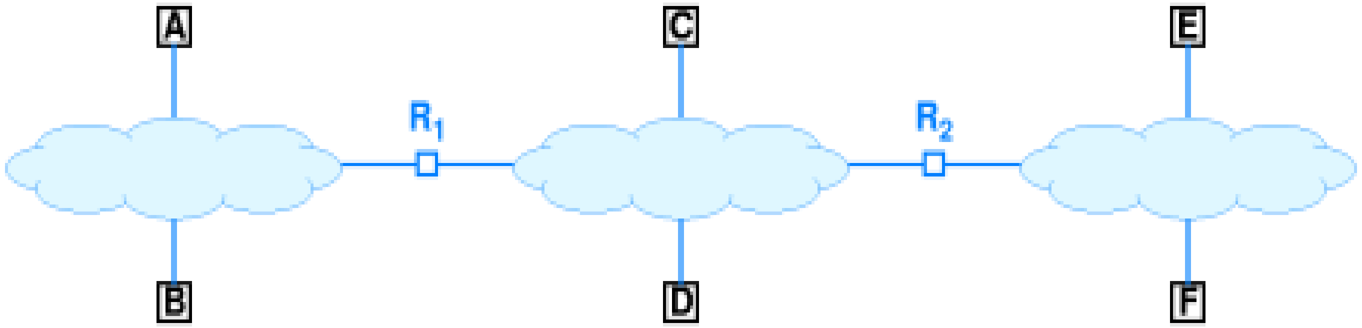
Resolving Addresses

- Hardware only recognizes MAC addresses
- IP only uses IP addresses
- Consequence: software needed to perform translation
  - Part of network interface
  - Known as *address resolution*

Address Resolution

- Layer 2 protocol
- Given
  - A locally-connected network, *N*
  - IP address *C* of computer on *N*
- Find
  - Hardware address for *C*
- Technique
  - Address Resolution Protocol

Address Resolution



Example

- When hosts A and B are on the same LAN, A wants to send a packet to B, and A has only the IP address of B, then A uses some method of *address resolution* to translate the IP address to the physical address.
- A can then send the message directly to B.
- If A and B are not on the same LAN, then A does not seek the physical address of B.
- A will send the packet to a router R1 across the local LAN or other link.
- R1 will forward the packet to get it close to F.
- If A does not know the physical address for R, A may use address resolution to obtain it.
- IP packets just get passed from router to router to router until they arrive at a router G connected to the destination network.

Address Resolution Techniques

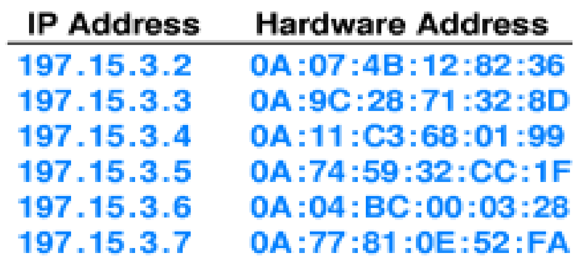
- What algorithm does software use to translate a protocol address into hardware address?
- The answer depends on the protocol and hardware addressing schemes.
- For example, the method used to resolve an IP address to an Ethernet address differs from the method used to resolve an IP address to an ATM address because the Ethernet addressing scheme differs from the ATM addressing scheme.

Address resolution algorithms

- "There's more than one way to skin a cat."
- Address resolution algorithms may be grouped into three basic categories:
- ***Table lookup***. Bindings or mappings are stored in a table in memory, which the software searches when it needs to resolve an address.
- ***Closed-form computation***. The protocol address assigned to a computer is chosen carefully so that the computer's hardware address can be computed from the protocol address using Boolean and arithmetic operations.
- ***Message exchange***. Computers exchange messages across a network to resolve an address. One computer sends a message that requests an address binding (i.e. translation), and another computer sends a reply that contains the requested information

Address Resolution with Table Lookup

- The table lookup approach to address resolution requires a data structure that contains information about address bindings.
- The table consists of an array. Each entry in the array contains a pair (P, H), where P is a protocol address and H is the equivalent hardware address



IP Address	Hardware Address
197.15.3.2	0A:07:4B:12:82:36
197.15.3.3	0A:9C:28:71:32:8D
197.15.3.4	0A:11:C3:68:01:99
197.15.3.5	0A:74:59:32:CC:1F
197.15.3.6	0A:04:BC:00:03:28
197.15.3.7	0A:77:81:0E:52:FA

Address Binding Table

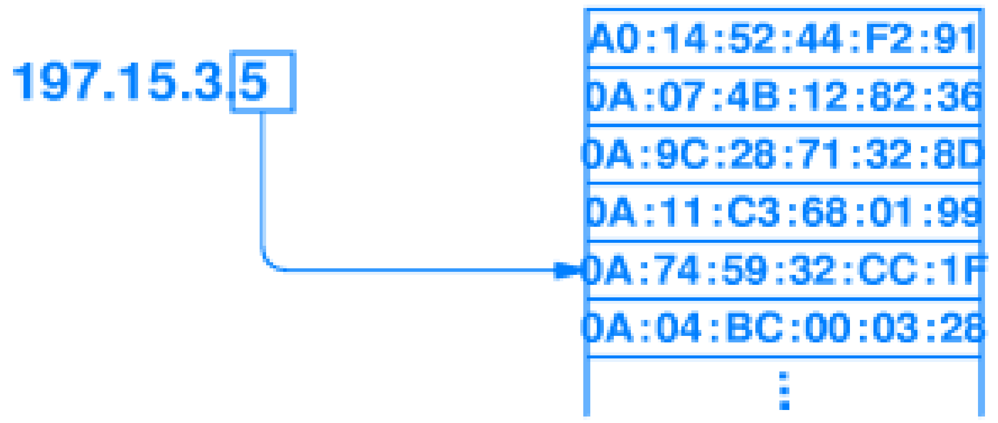
- A separate address binding table is used for each physical network.
- Consequently, all IP addresses in a given table have the same prefix.
- For example, the address binding table in the figure corresponds to a network with the class C number 197.15.3.0.
- Therefore, each IP address in the table will begin with the 197.15.3 prefix.
- Implementations can save space by omitting the prefix from table entries.

Table Lookup

- The basic idea of table lookup is just to look up the IP address in a table (often implemented with an array) and read off the corresponding physical address.

  If the IP addresses are all sequential and the table is laid out in order of increasing IP address, then software can be written so it "jumps" to the correct entry in the table immediately.
- If not, there are programming tricks (e.g. hashing) that can simulate the situation where the numbers are sequential.
- This is important because it can take too long to search a long table for an IP address.

Illustration



Advantage Of The Table Lookup

- The chief advantage of the table lookup approach is generality - a table can store the address bindings for an arbitrary set of computers on a given network.
- In particular, a protocol address can map to an arbitrary hardware address.
- Furthermore, the table lookup algorithm for address resolution is straightforward and among the easiest to program.
- Given a next-hop IP address, N, the software searches the table until it finds an entry where the IP address matches N. The software then extracts the hardware address from the entry.

Address Resolution With Closed-Form Computation

- Closed-form computation calculates the unknown MAC address from the known IP address.
- The sending node fills in the destination MAC in the Ethernet frame from the calculated value.
- This method is very quick and does not require outside resources or communication.
- It also allows reasonably tight control over the address space.
- However, it does require configurable MAC addresses and some level of management, as the addresses must all be assigned to the various hosts.

Closed-Form Computation

- It may be possible to assign the physical addresses and IP addresses in such a way that there will be a simple arithmetic formula that can be used to translate from one address to the other.

  As a simple example, Ethernet addresses are 6 byte numbers and IP addresses are 4 byte numbers. The Ethernet address corresponding to A.B.C.D could be 0.0.A.B.C.D.

  The closed-form method is very "nice" because it is efficient in its use of both time and memory.

CFC

- To understand why closed-form computation can be especially efficient for a network with configurable addresses, remember that both the hardware and IP addresses can be changed.
- Thus, values can be chosen to optimise the translation.
- In fact, the host portion of a computer's IP address can be chosen to be identical to the computer's hardware address, making the translation trivial

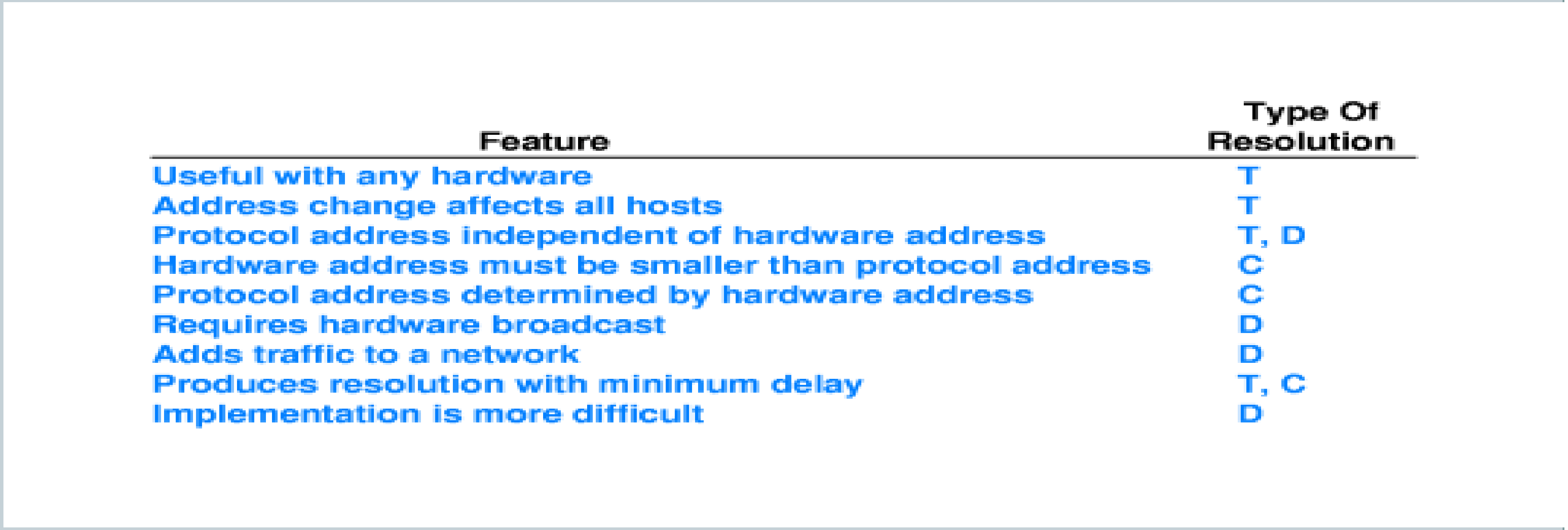
Example

- As an example, suppose a configurable network has been assigned the class C network number 220.123.5.0.
- As computers are added to the network, each computer is assigned an IP address suffix and a matching hardware address.
- The first host is assigned IP address 220.123.5.1 and hardware address 1.
- The second host is assigned IP address 220.123.5.2 and hardware address 2.
- The suffixes need not be sequential.
- If a router attached to the network is assigned IP address 220.123.5.101, the router is assigned hardware address 101.
- Given the IP address of any computer on the network, the computer's hardware address can be computed by a single Boolean *and* operation:
-
- Hardware_address = ip_address & 0xff

Address Resolution With Message Exchange

- Hosts can send messages to an address resolution *server*.
- The servers can be a problem because their tables have to be kept current and they can be swamped with requests.
- Hosts can broadcast a request for address resolution. The intended recipient recognizes its own IP address sent out in the broadcast and replies by sending its hardware address to the host that made the broadcast.
- This distributed method can be a problem because broadcasts force all hosts to use their CPU's.
- On an Ethernet the method is usually message exchange, except that results are cached in a table.

Summary



Feature	Type Of Resolution
Useful with any hardware	T
Address change affects all hosts	T
Protocol address independent of hardware address	T, D
Hardware address must be smaller than protocol address	C
Protocol address determined by hardware address	C
Requires hardware broadcast	D
Adds traffic to a network	D
Produces resolution with minimum delay	T, C
Implementation is more difficult	D

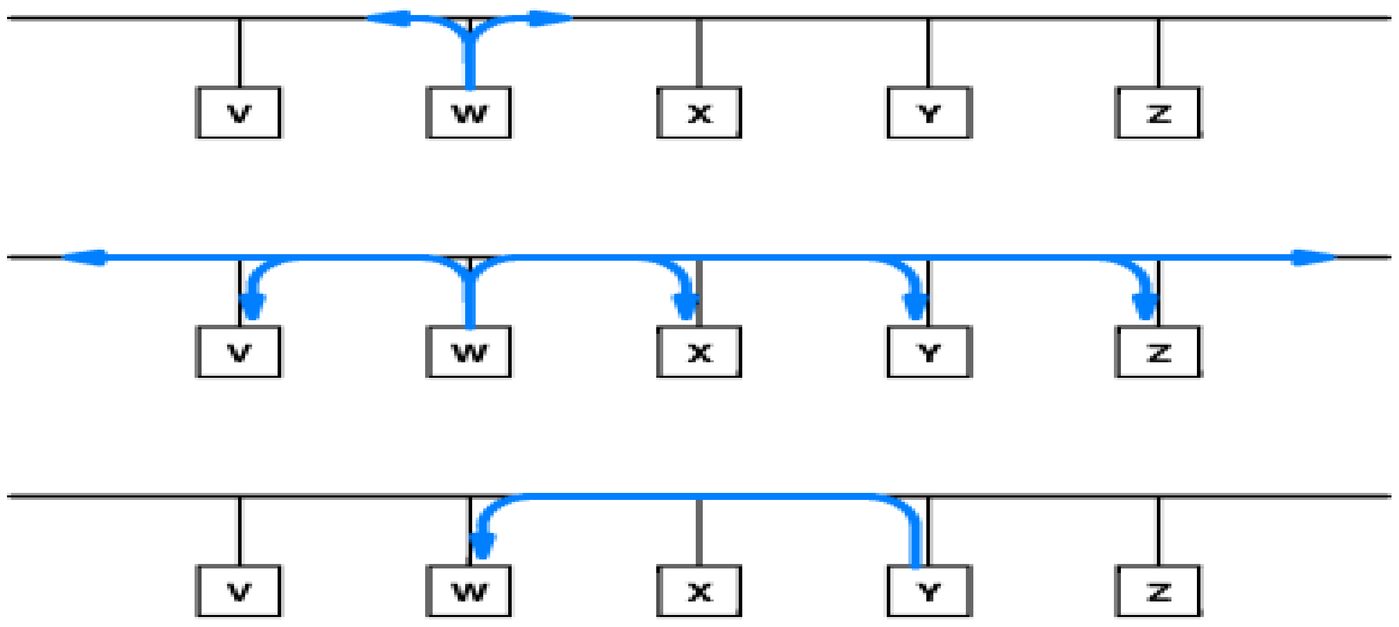
Address Resolution Protocol

- The TCP/IP standard does not dictate what method of address resolution to use.
- Generally, TCP/IP uses table lookup to find hardware addresses for WAN links.
- Address Resolution Protocol (ARP) is a TCP/IP protocol/standard for message-based address resolution.
- TCP/IP commonly uses ARP on LAN's with static hardware addressing.

ARP Message Delivery

- Suppose computer W wishes to send a packet to computer Y. The protocol requires that the ARP request be broadcast on the local LAN in a single packet containing the IP address of W, the hardware address of W, and the IP address of Y.
- Then Y is supposed to respond to W.
- All other hosts process the request, but make no reply.
- Y does not broadcast. It simply places its hardware address in a packet and transmits it to W.

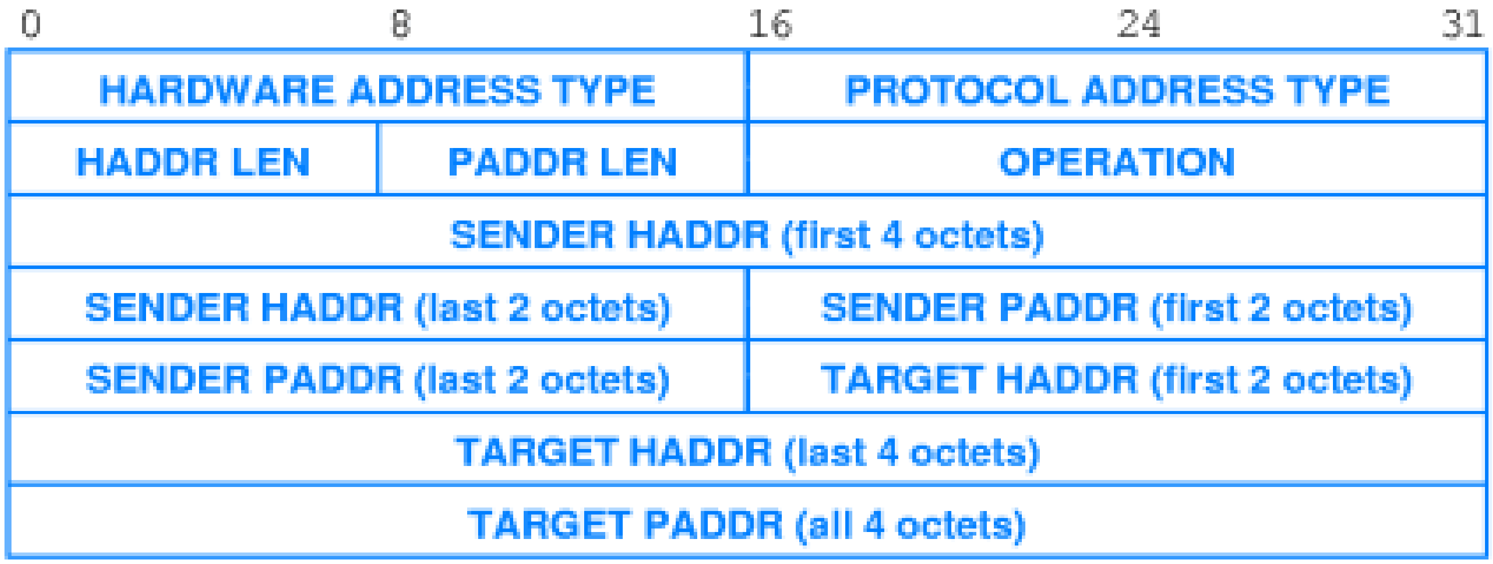
Example



ARP Message Protocol

- A field in the ARP packet tells the length (in bytes) of a protocol address and the length of a physical address.
- Usually the protocol address is an IP address and the physical address is an Ethernet address, but ARP can be used to translate between arbitrary protocol and hardware addresses.
- The fields of the ARP packet are shown in the figure below,

Message format



0 8 16 24 31

HARDWARE ADDRESS TYPE		PROTOCOL ADDRESS TYPE	
HADDR LEN	PADDR LEN	OPERATION	
SENDER HADDR (first 4 octets)			
SENDER HADDR (last 2 octets)		SENDER PADDR (first 2 octets)	
SENDER PADDR (last 2 octets)		TARGET HADDR (first 2 octets)	
TARGET HADDR (last 4 octets)			
TARGET PADDR (all 4 octets)			

Format Explanation 1

- The first two 16-bit fields contain values that specify the type of hardware and protocol addresses being used.
- For example, field HARDWARE ADDRESS TYPE contains 1 when ARP is used with Ethernet, and field PROTOCOL ADDRESS TYPE contains 0x0800 when ARP is used with IP.
- The second pair of fields, HADDR LEN and PADDR LEN specify the number of octets in a hardware address and protocol address. Field OPERATION specifies whether the message is a request (value 1) or a response (value 2).

Format explanation 2

- Each ARP message contains fields for two address bindings. One binding corresponds to the sender, while the other corresponds to the intended recipient, which ARP calls the *target*.
- When a request is sent, the sender does not know the target's hardware address (that is the information being requested for).
- Therefore, field TARGET HADDR in an ARP request can be filled with zeroes because the contents are not used.
- In a response, the target binding refers to the initial computer that sent the request.
- Thus, the target information in a response serves no purpose - it has survived from an early version of the protocol.

Sending an ARP Message

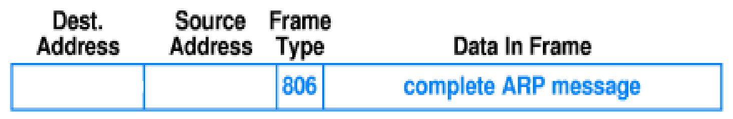
- A host sending an ARP packet places it inside a physical frame (just as any other "data" would be).
- In other words the ARP packet is encapsulated in a hardware frame and treated as "freight" for its trip across the physical network.
- The ARP message is treated as data being transported - the network hardware does not know about the ARP message format and does not examine the contents of individual fields.
- Technically, placing a message inside a frame for transport is called *encapsulation*.
- ARP is encapsulated directly in a hardware frame
-

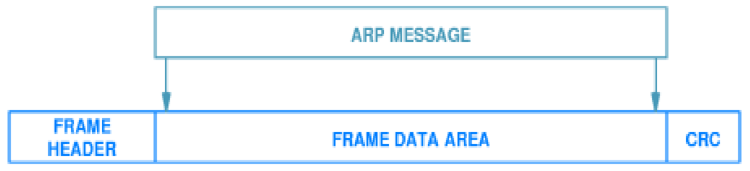
Identifying ARP Frames

- The hardware frame has a type field.

  The sender of an ARP packet places a special code in the type field (0x806 for an Ethernet frame carrying an ARP packet) so the receiver will know that the data portion of the packet contains an ARP message.

  This is useful to the receiver, to help it determine to what protocol stack or software module the packet should be given.





Caching ARP Responses

- A host W that sends an ARP request will get a reply containing a *binding*. W will *cache* the binding in a table.

  That way, W may not have to broadcast another ARP request the next time it needs to resolve that same IP address -- a waste of network bandwidth and CPU time on all the local hosts.

  A host must *age* and eventually *expire* information in its ARP cache, else it would keep obsolete information indefinitely.

  ARP information has to change: IP address assignments can change and host NIC's can be replaced.

Processing An Incoming ARP Message

- Suppose a computer W broadcasts an ARP request on a LAN, in an attempt to learn the physical address of a computer Y.
- Of course, the ARP request is delivered to *all* the computers on the LAN. Suppose Z ≠Y is a computer on the LAN that already has a version of W's binding in its cache.
- Interestingly, Z will update its cache with the binding for W contained in the ARP request.

  This is useful because, since Z currently has a binding for W, it has probably been communicating with W in the recent past.
- It will probably need to communicate with W in the near future. If Z takes the opportunity now to update its binding for W, then maybe that will save Z some time and effort. Z won't have to send an ARP request to W in the near future.

Continuation

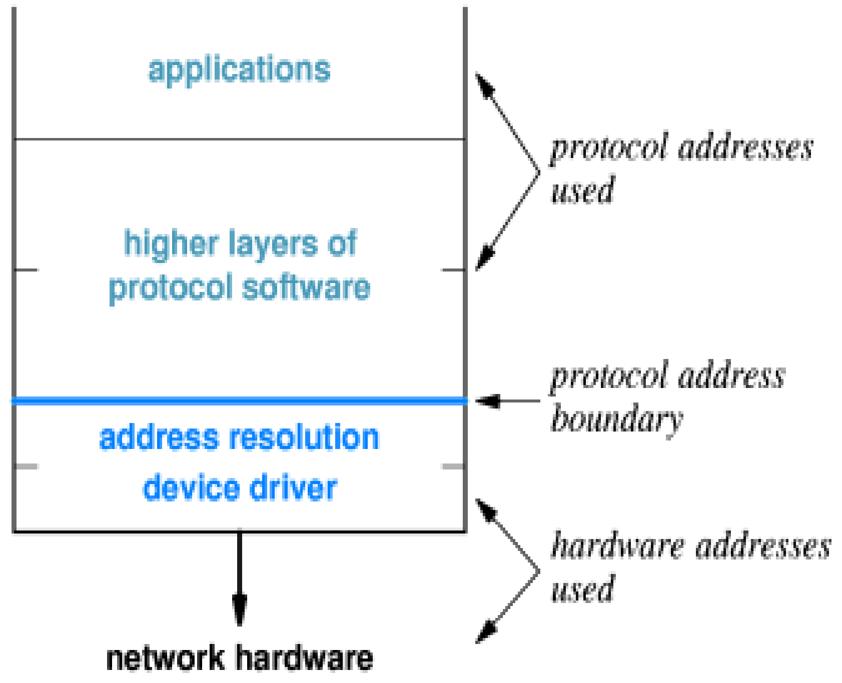
- Z updates its cache with W's binding *only if* Z's cache *already* has a version of W's binding.

  Y, the target of the ARP request from W, behaves differently. Y places W's binding in its cache *whether or not* it already has binding for W.

  This is worthwhile because W is almost sure to send a packet to Y immediately and Y is almost sure to reply to W. Anticipating this, Y caches W's binding.

Layering, Address Resolution, Protocol Addresses

- ARP operates at the "Network Interface Layer" of the TCP/IP protocol stack.



Summary

- *Because ARP software is part of the network interface software, all higher-layer protocols and applications can use IP addresses exclusively, and remain completely unaware of hardware addresses.*