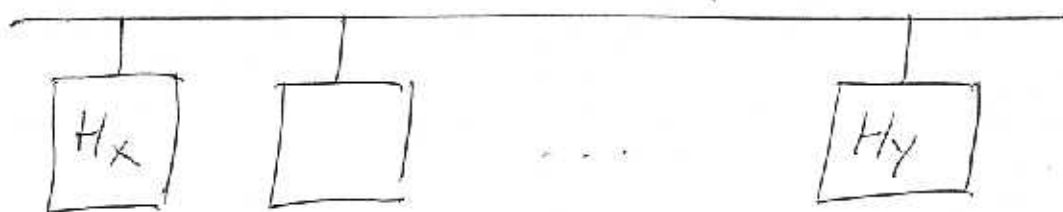


Start here 01/30/2004

(From: ~~Packet~~ Addresses, p 20) No, later

30.



H_x has IP packet For H_y

knows: (1) Same Network

(2) IP Address H_y

Q: (pp 12-14)

How does H_x know the
physical address of H_y ?

Answer: ARP

Address Resolution Protocol

Every Computer (H_x)
has an ARP table

IP address	Physical address
entry 1	
entry 2	
etc	

where for some of the (other) computers on the same network it (interfaces) maps IP Address into Physical Address.

Why not all ?

Why not all?

To accommodate changes.

Hx has IP address,
needs physical address.

① check ARP Table. IF there: Ok.

② IF Not:

Send ARP Request

Comer, pp 84-85

Broadcast.

Physical Broadcast:

Dest.
Address

FF: FF: FF: FF: FF: FF

(48 ones: F = 15 = 1111)

Includes IP address for which
physical address sought.
& own logical address, physical address.

Broadcast: Every body Listens.

33

• For ^{H_y:} me? if yes:

Send ARP Reply. Unicast.
(Physical Unicast).

• Back to H_x:

Updates ARP Table, sends packet.

• Next time:

There is an entry in the
ARP Table of H_x For H_y:

No ARP Request needed.

Does this work?

What if H_y changes
physical address?

Soft State :

(Time Out).

Theory : Every Row in the ARP Table has a time-out stamp: Valid until.

State	Queue	Attempt	T.O.	Prot. Addr (IP)	Hardware Addr (physical)

State:
 / Free (entry expired)
 — Pending (Request has been sent, no response yet)
 \ Resolved (OK)

Queue: (IF Pending) identity of Queue where packet(s) are waiting.

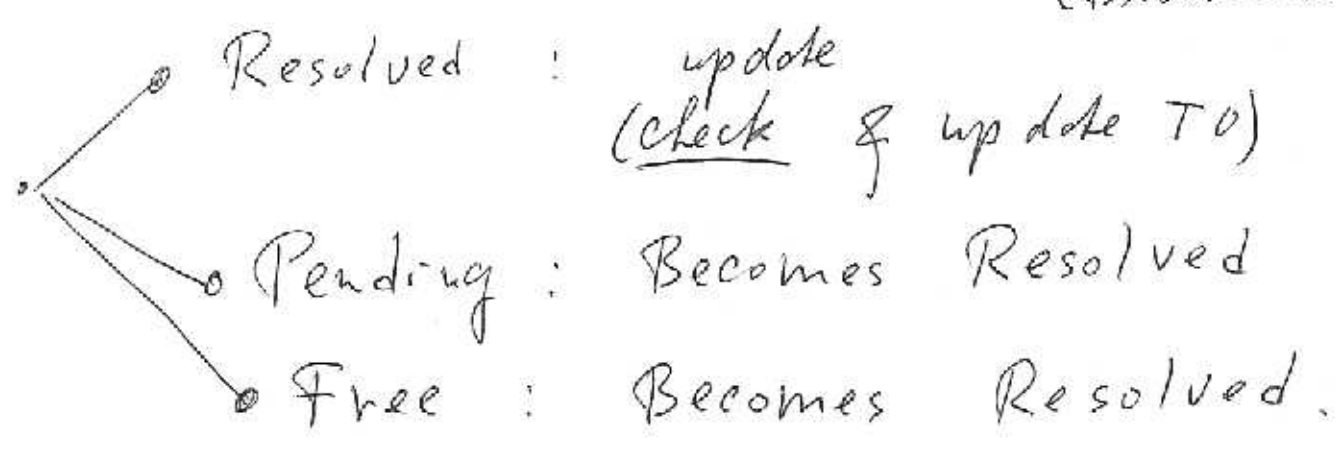
Attempts: Time-out Field For "Pending" increases ever (5?) sec, until TO. Re-sends at increase.

TO: Time-Out Field For "Resolved".

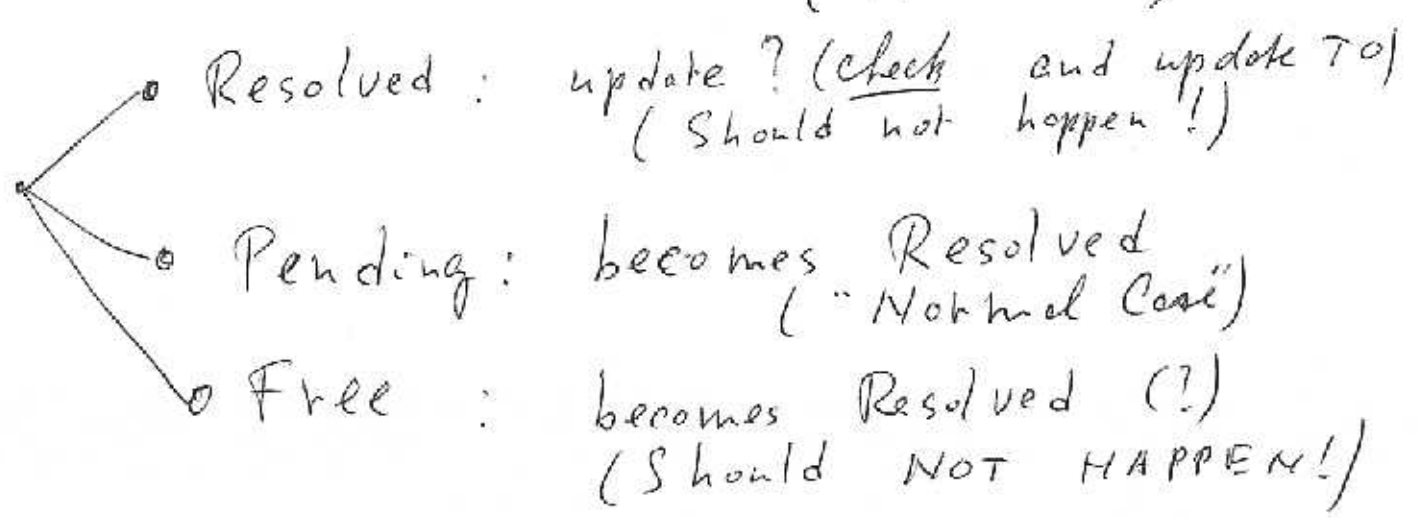
(IF "TO" Reached: now becomes "FREE").

(In OS 5 and up: it is different!)

IF host sees ARP-Request: (Broadcast!)

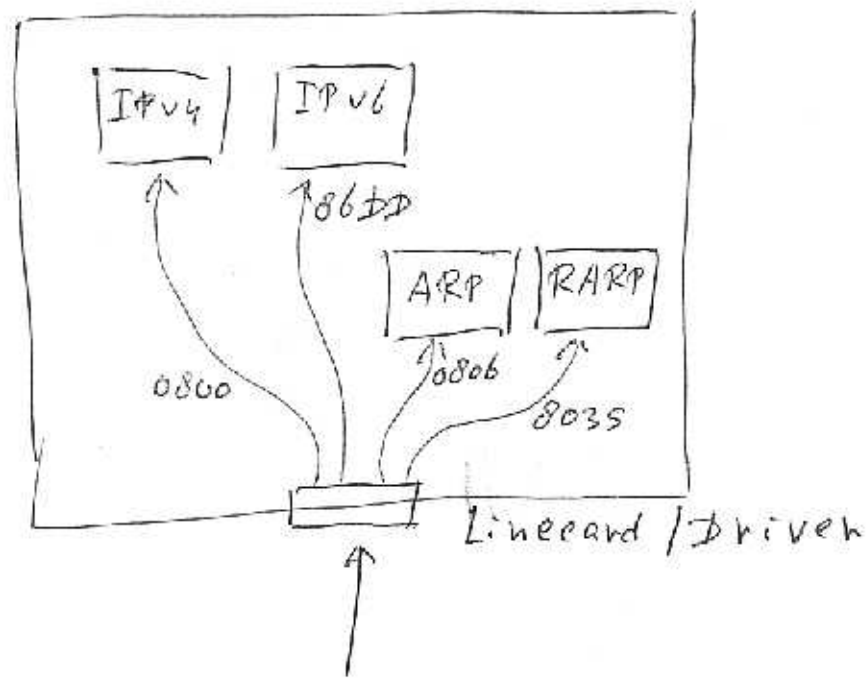


IF host sees ARP-Reply: (Unicast!)



Ethertype:

- IPv4 : 0800 (Hexadec)
- IPv6 : 86DD
- ARP : 0806
- RARP : 8035 (later)



Ethertypes (Frame Types)
ARE necessary!

ARP, RARP packets are NOT
 IP packets! other Format!

RARP

Reverse Address Resolution
Protocol.

Comer, Ch. 6. (pp 89-93)

RARP has largely been
replaced by BOOTP and
DHCP, Comer Ch. 23 (later).

RARP

A diskless workstation, at boot, does
not know its own IP address.

But it knows (can find) its own
physical address.

Sends "RARP Request".

RARP Request.

38

"This is my physical address.
please send me my IP address."

There must be a

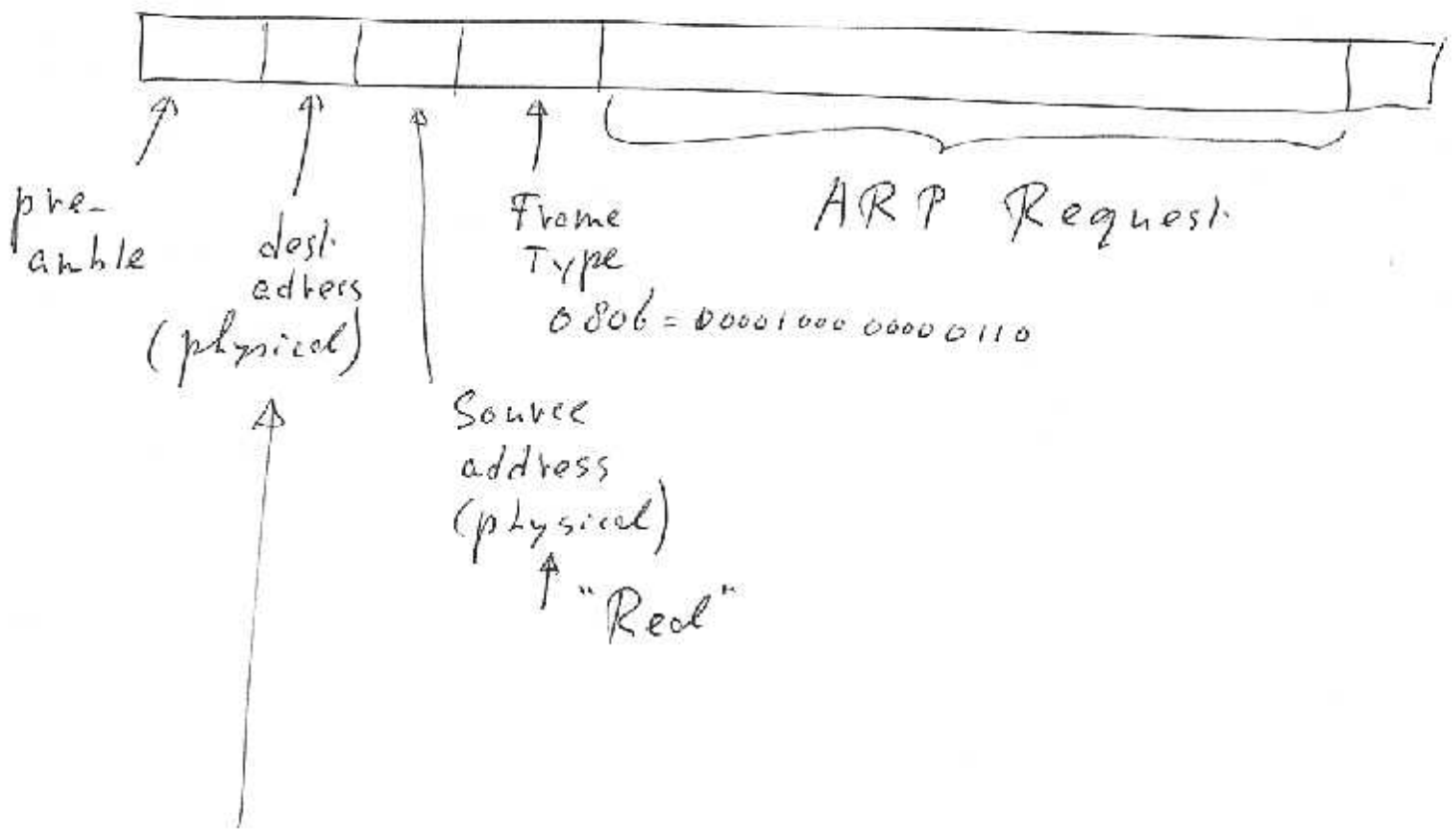
"RARP Server" on the
LAN (?)

Takes the request.

Sends Response:

RARP Response.

ARP Request, Layout (ethernet)



FF: FF: FF: FF: FF: FF.

The physical broadcast address.

Question: What ~~is~~ are the
IP source address, IP destination
addresses in the Header of the
IP Packet?

Answer: No such things!

This was a trick question.

40

The ARP Request is NOT an IP packet!

(Hence : different Frame Type).

See p. 36 of these notes.

The

Linecard / ethernet card / NIC / driver uses the ethertype or frametype of a frame (of a LAN frame, ethernet frame, link layer frame) to decide which process to send the packet to.

Layout of ARP/RARP packet. 41

Hardware Type ⁽¹⁶⁾		Protocol Type ⁽¹⁶⁾
HLEN ⁽⁸⁾	PLEN ⁽⁸⁾	Operation ⁽¹⁰⁾
Sender hardware address: as long as needed!		
Sender protocol address: as long as needed!		
Target hardware address: as long as needed!		
Target protocol address: as long as needed!		

Target!
not dest!

Hardware Type: describes Type of LAN
 For ethernet: 1 (0000 0000 0000 0001)

Protocol Type: describes which Higher level protocol
 For IPV4: 0800
 (Same as ethernet).

HLEN: Length of hardware address. in Bytes
 ethernet: 6 = 0000 0110

PLEN: Length of protocol address: in Bytes.
 IPV4: 4 = 000 0100

OPER: ARP Request = 1 (0000 0000 0000 0001)
 ARP Reply = 2
 RARP Request = 3
 RARP Reply = 3

The "Sender Hardware Address field"
is "HLEN" Bytes long.

Where can you find this kind of
information?

Of Course : www.iana.com

Protocol Number Assignment Services

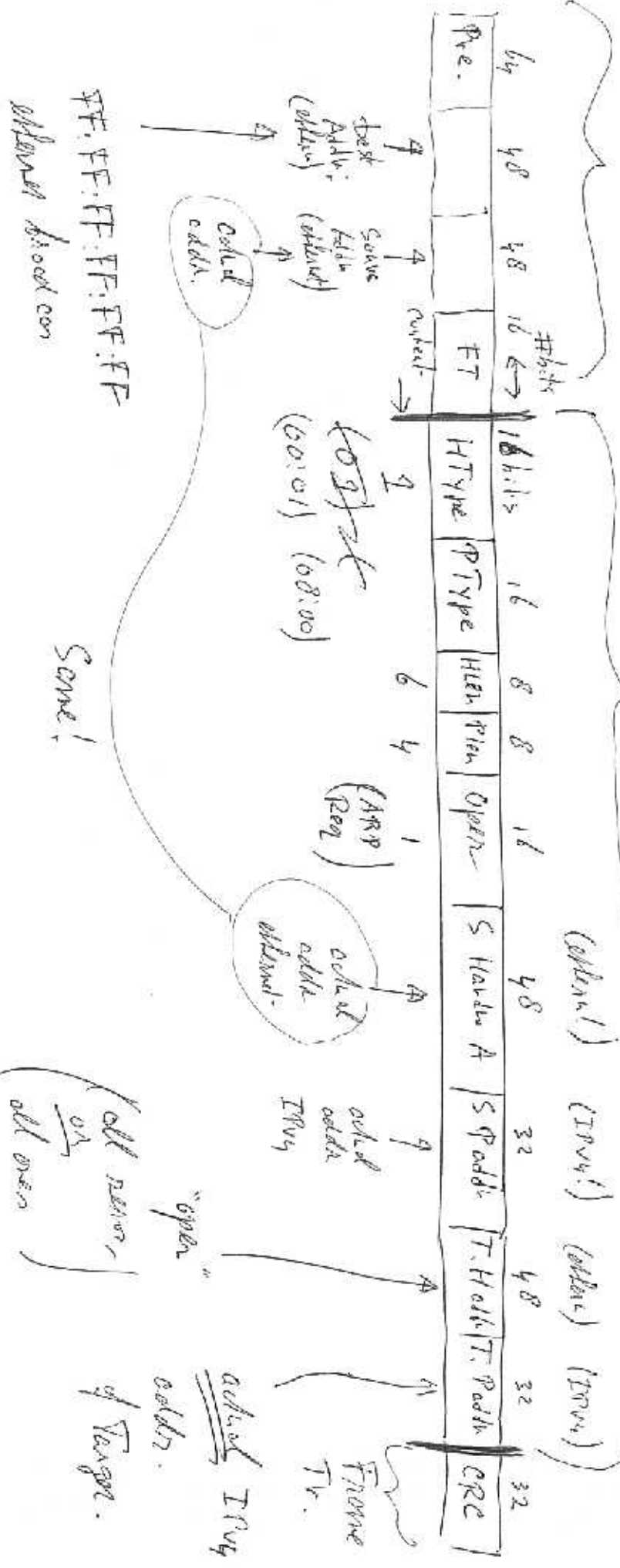
A

Address Family Numbers

ARP Parameters

There will be homework on this for
02/10/2004

ARP Request Packet inside ethernet frame for IPv4 Protocol



("should" be none!
But I have seen
all ones)

Note:

In the frame header we talk about source and destination.

In the ARP Packet ~~we~~ we talk about source sender and target (not dest).