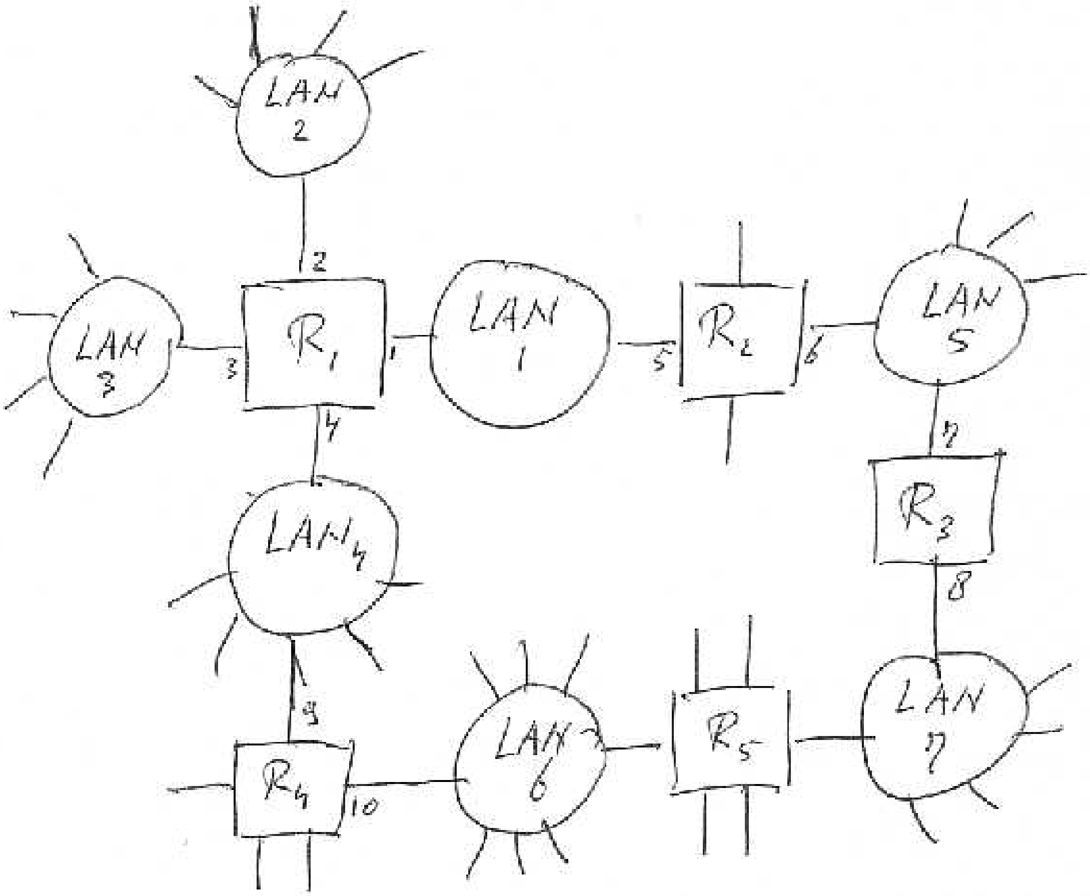
At this point we "kind of finished"
Ch 1-6 (pp 1-94)
(But there are some loose ends).
Before we start on Ch 7
(Internet packet, header, datagram) a
small detour:
<u>Forwarding</u>.
What Comer (Ch 8) calls <u>Routing</u>
really is <u>Forwarding</u>.
<u>Routing</u> will be done in Comer
Ch 14, 15, 16, and a bit in Ch 17.



R_1 gets, on interface 3, a LAN-frame with an <u>Internet Packet</u> inside.

<u>How</u> does R_1 decide what to do with this packet?

(1) The source of the packet may know everything and include the whole route in the packet.
("Strict Source Route").

(2) (Practically Always)
R_1 has a forwarding table.
"what do I do next". (or "now").

Forwarding Table.
Comer calls this a routing table (e.g. p 120)

Oversimplified.

Oversimplified Forwarding Table.
in R_1

Netw	Action	Interface (name)	Next Hop (IP addr)
LAN1	DD	1	
LAN2	DD	2	
LAN3	DD	3	
LAN4	DD	4	
LAN5	Forw	1	5
LAN6	Forw	4	9
LAN7	? Forw	?	?

Routing decision

Forwarding: Use the forwarding table.
Routing: Make the forwarding table.

The forwarding table on p 48 requires that R_1 can decide ~~the~~ what the ultimate destination LAN of a packet is.

With classfull addressing that works.

With "the other" method: (CIDR notation)

R_1 needs a way to decide what the ultimate network is.

---

128 192 224

CIDR Notation.

128.235.32.0/22 :

The set of all addresses that have the same first 22 bits as 128.235.32.0.

$\underbrace{1000\ 0000\ 1110\ 1011\ 0010\ 00}_{22 \text{ bits}} \underbrace{\text{anything}}_{10 \text{ bits}}$

Mask Notation:

Network Address: 128.235.32.0

1000 0000 1110 1011 0010 00 00 0000 0000

Mask 1111 1111 1111 1111 1111 11 00 0000 0000

(22 ones, followed by 32-22=10 zeros)

(Same as 128.235.32.0/22)

"All addresses that if logically ANDed with the mask give the network address".

(The network address must have the last 10 bits zero!)

Forwarding Table. few Oversimplified

Mask	Netw	Action	Interface (Name)	Next Hop (IP addr)	Me
----	----	---	--	--	--

Take destination address.
Logical AND with mask i
if result identical to network i:
"route i".

Masks: "contiguous".
a few ~~xx~~ ones, followed by only zeros.

What if the mask is
"all ones"? (255.255.255.255)?

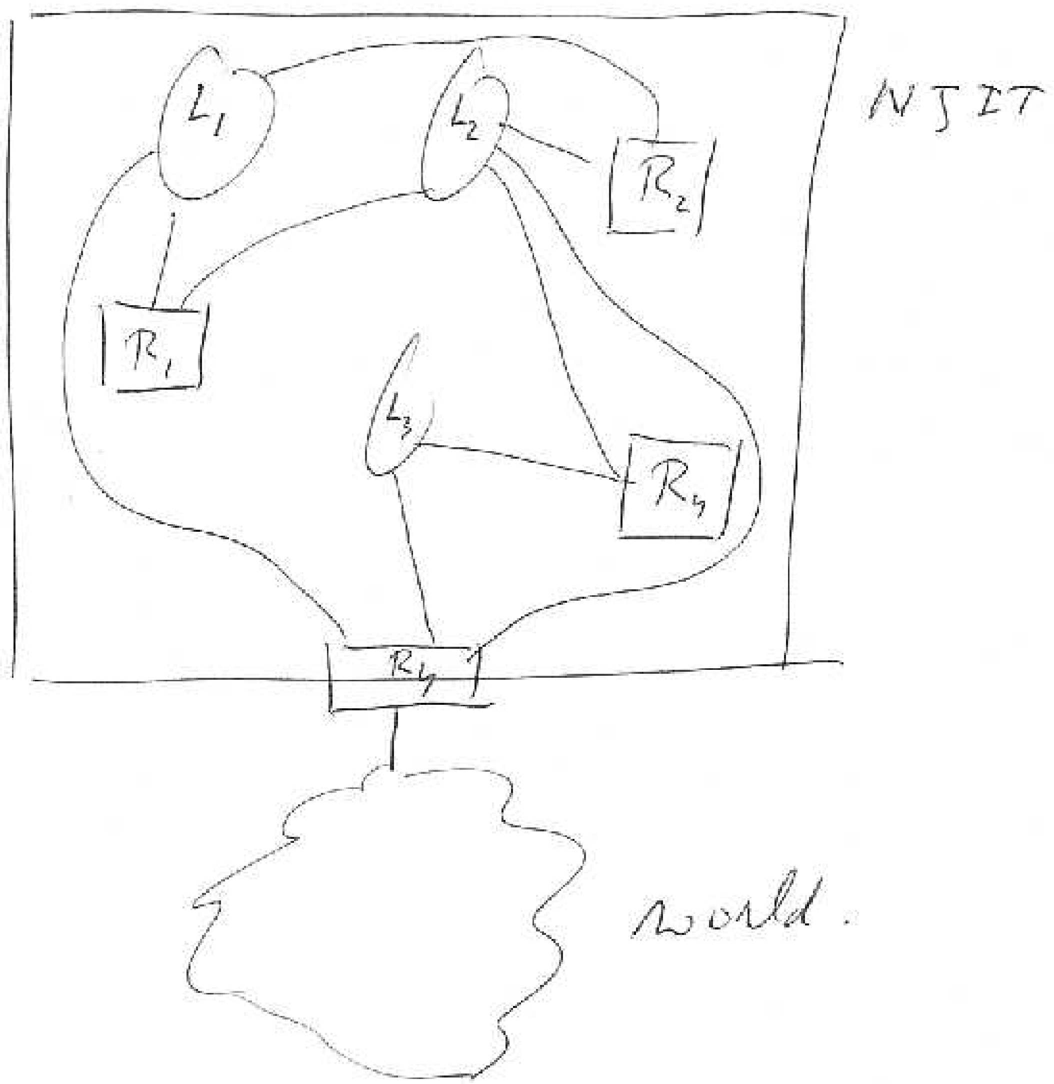
destination address must be <u>identical to</u>
network address!

"Host-specific Route".

Now we know what the <u>mask</u> in
the ARP table means!
(arp -a, or arp -na, or arp -an).

~~The above assumes~~
~~"Exactly one fit".~~

In addition, many routers have a
"default route".
If no route fits, do this: ... .

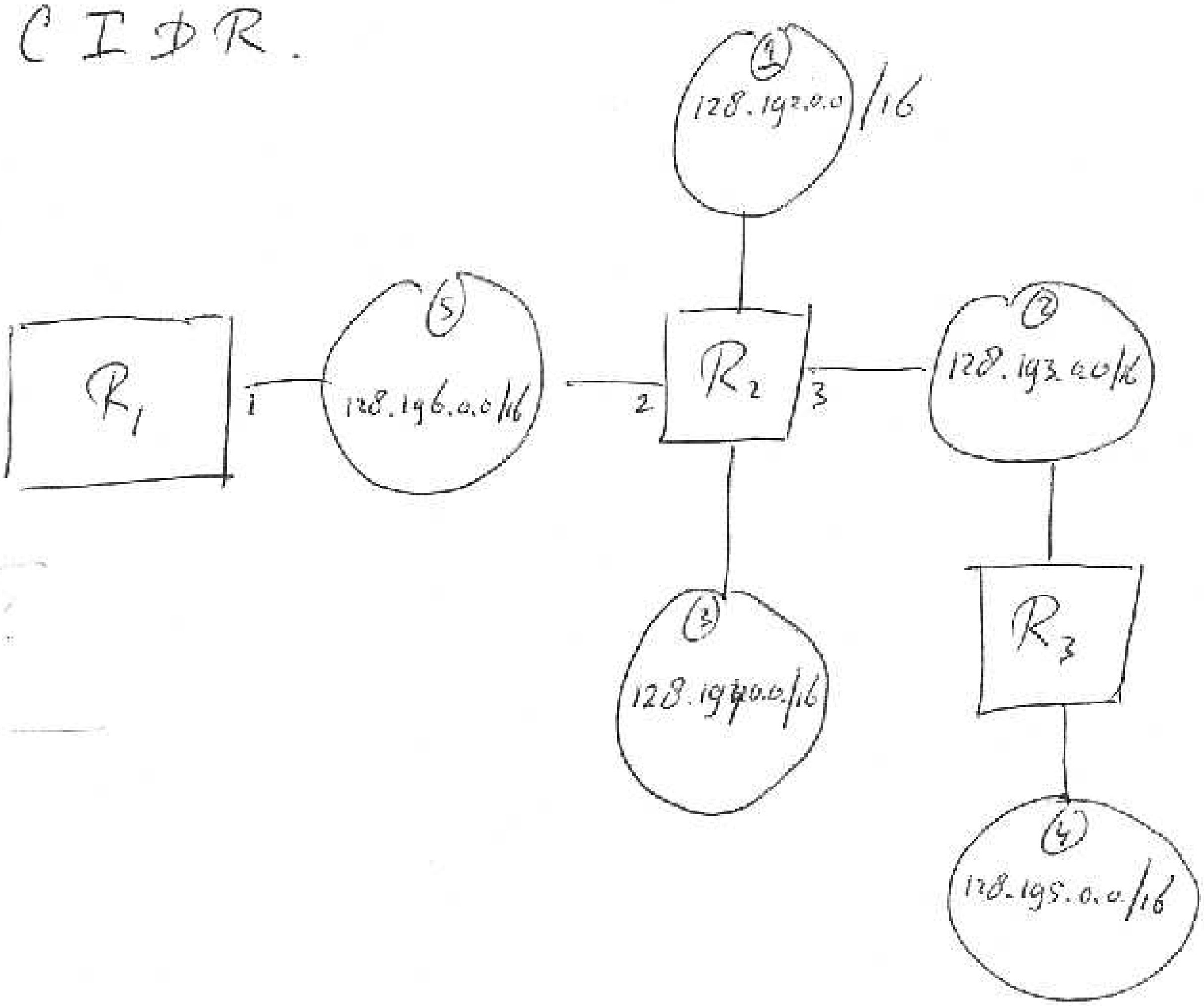


R_1 has routes to L_1, L_2 (DD)
and to L_3 (via any R)
and to "rest" via R_4 : ~~defut~~ default.

Only Routers "in the periphery" have default routes!
(Suppose all routers had a default route!).

Thus far we assumed every dest. addr. had, in the forwarding table of R, at most one fit.
(This "ought" to be the case).
But: (later).

CIDR.



(1): 1000 0000 1100 0000 { anything
(2): 1000 0000 1100 0001 { anything
(3): 1000 0000 1100 0010 { anything
(4): 1000 0000 1100 0011 { anything

Can be ~~cont~~ combined !

(5): 1000 0000 1100 0100 { anything

R_1 could have:

Mask	Netw	Action	Interf	Next Hop
"16"	128.192.0.0	F	1	2
"16"	128.193.0.0	F	1	2
"16"	128.194.0.0	F	1	2
"16"	128.195.0.0	F	1	2
"16"	128.196.0.0	DD	1	

Combine

but R has the right to combine the first 4:

"16" = 255.255.0.0
"14" = 255.252.0.0

"14"	128.192.0.0	F	1	2
"16"	128.196.0.0	DD	1	

Saves space in the forwarding table

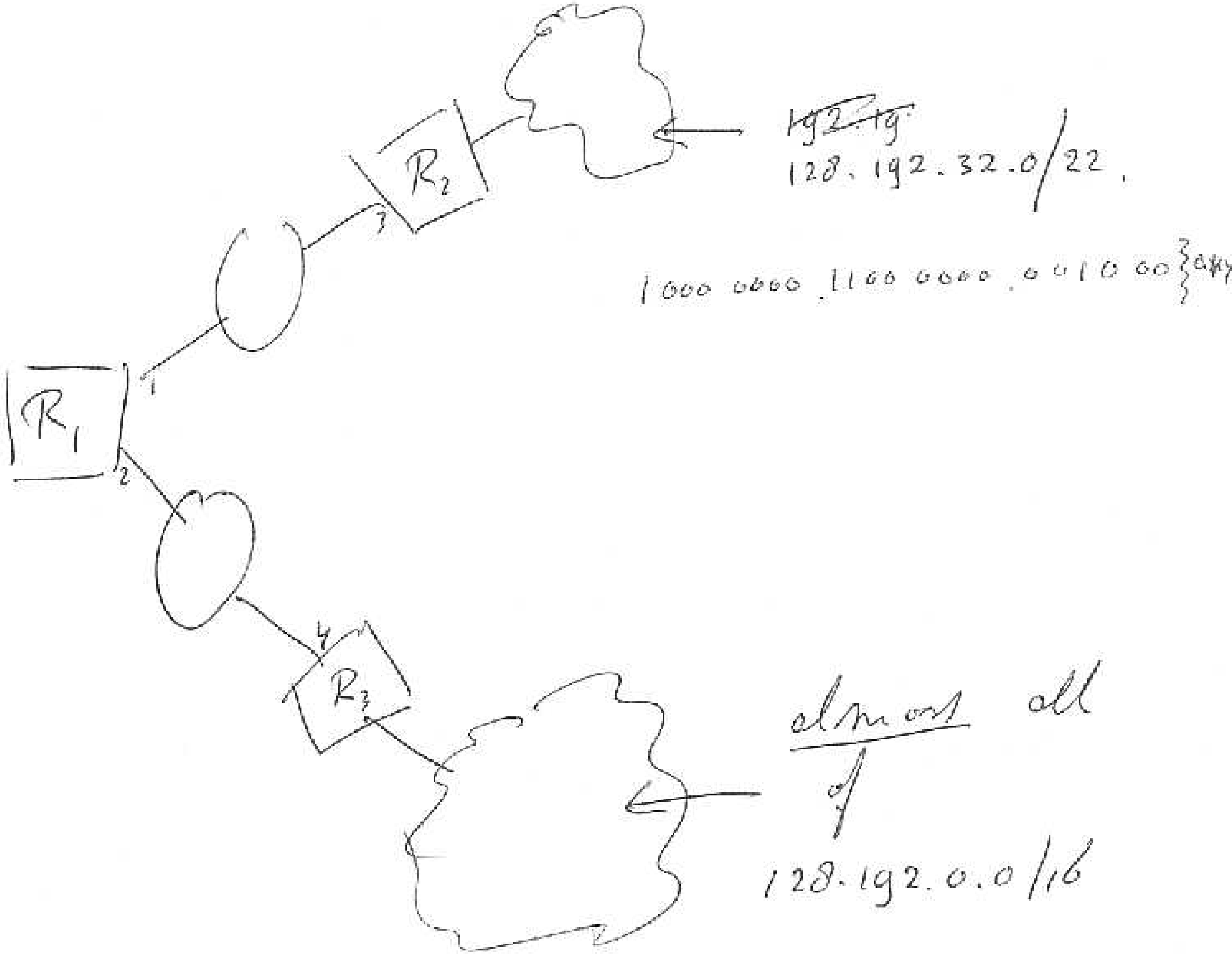
Must be "contiguous" and "the right boundaries". We can do this collapse in the Forwarding Table. NOT in the Routing Table!

We <u>would prefer</u> that every destination address has a fit with <u>at most one</u> (exactly one(?)) route in the forwarding table.

Sometimes you can "almost" collapse a number of routes into one, with just one or two exceptions.

Then you <u>may</u> do it.

<u>Longest Prefix Routing</u>.



Now we may decide to put into the forwarding table of R_1: