

Net_practice

In this project we learn the basics of networking, learning what are the main technologies, protocols and devices that go into making networks operational and stuff..

The actual project is solving a series of exercises where we must connect different devices to one another over a network, solving the problems isn't that hard and we can always ask for help but really understanding why a given solution works and even more important knowing all the possible solution and all scenarios where it wouldn't work that is, to me, the real project.

Network_devices

HUB – very dumb and obsolete

In networking, a hub is a device that links multiple computers and devices together. Hubs can also be referred to as repeaters or concentrators, and they serve as the center of a local area network (LAN). In a hub, each connected device is on the same sub-net and receives all data sent to the hub. The hub then forwards that data out to all other connected devices.



The most basic network is just two PCs connected via an Ethernet cable, forming a **peer-to-peer** connection. When we add a **hub**, we make it possible to connect more PCs, but the way a hub handles data makes it highly inefficient.

A hub simply **broadcasts** any data it receives to **all connected devices**, regardless of the intended recipient. This leads to:

- **High collision rates** - Since all devices share the same bandwidth, simultaneous transmissions cause data collisions, requiring retransmissions.
- **Lower performance** - As more devices join, the network slows down due to constant collisions and retransmissions.
- **No security or filtering** - Any device connected to the hub receives all traffic, making it easy to intercept data.

Because of these issues, **hubs are now obsolete** and have been replaced by **switches**, which intelligently direct data only to the intended recipient, eliminating collisions and improving speed and efficiency.

Most "hubs" sold today are actually **switches** in disguise, as true hubs are rarely used in modern networks. 🚀

Switch - allows us to make networks

Is networking hardware that connects devices on a computer network by using packet switching to receive and forward data to the destination device. A network switch is a multi-port network bridge that uses MAC addresses to forward data at the data link layer (layer 2) of the OSI model. Some switches can also forward data at the network layer (layer 3) by additionally incorporating routing functionality. Such switches are commonly known as layer-3 switches or multilayer switches.



Router – routes stuff

A router is a computer and networking device that forwards data packets between computer networks, including inter-networks such as the global Internet.

Routers perform the "traffic directing" functions on the Internet. A router is connected to two or more data lines from different IP networks. When a data packet comes in on a line, the router reads the network address information in the packet header to determine the ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Data packets are forwarded from one router to another through an inter-network until it reaches its destination node.[5]

TCP/IP

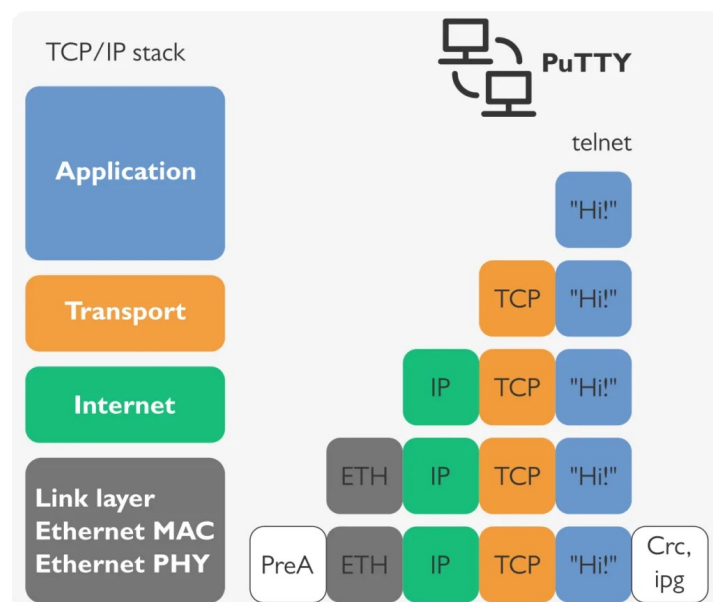
TCP/IP is a set of rules (protocols) that allows computers to communicate with each other over a network. It's like the "language" devices use to send and receive data reliably and efficiently.

TCP (Transmission Control Protocol): Handles breaking data into packets, sending them, and ensuring they arrive in the right order and without errors.

IP (Internet Protocol): Handles addressing and routing the packets to the correct destination.

Together, they make sure your data (like a webpage or an email) gets from point A to point B.

How Does TCP/IP Work?



Think of TCP/IP like sending a letter through the mail:

TCP is like tearing the letter into pieces (packets), numbering them, and ensuring they're reassembled correctly at the destination.

IP is like the postal service figuring out the address and delivering the pieces to the right mailbox.

Here's the step-by-step:

Your computer breaks data into small chunks called packets.

Each packet gets a header with info like the source IP address (your device), destination IP address (the receiver), and sequence number.

The packets travel across the network, possibly taking different routes.

At the destination, TCP checks that all packets arrived, puts them in order, and reassembles the original data.

Key Concepts

IP Addresses:

Every device on a network has a unique address, like a phone number. For example: 192.168.1.1.

Two versions exist:

IPv4: Older, uses 32 bits (e.g., 192.168.0.1), limited to ~4.3 billion addresses.

IPv6: Newer, uses 128 bits (e.g., 2001:0db8::8a2e:0370:7334), supports way more devices.

Ports:

Ports are like apartment numbers at the IP address "building." They tell the device which application (e.g., web browser, email) the data is for.

Examples: Port 80 (HTTP for websites), Port 443 (HTTPS), Port 25 (SMTP for email).

Layers of TCP/IP: TCP/IP is often explained using a 4-layer model:

1. Application Layer: Where apps like browsers or email clients live (e.g., HTTP, FTP).
2. Transport Layer: TCP (reliable) or UDP (faster, less reliable) manage data delivery.
3. Internet Layer: IP handles addressing and routing.
4. Link Layer: Physical connection (e.g., Ethernet, Wi-Fi).

Handshakes:

TCP uses a "three-way handshake" to start a connection:

Sender says, "Hey, let's connect!" (SYN).

Receiver says, "Cool, I'm in!" (SYN-ACK).

Sender confirms, "Great, we're connected!" (ACK).

Why It Matters

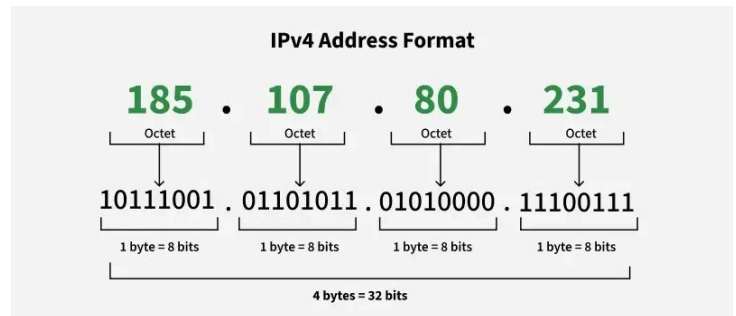
Reliability: TCP ensures no data is lost or corrupted (great for emails or file downloads).

Flexibility: IP works with any network type (Wi-Fi, Ethernet, etc.).

Scalability: It powers everything from your home router to the global internet.

OK so how to solve the problems!

The first lvls are pretty simple we just have a few pcs and maybe a switch, so just a bunch of mostly direct connections that need to be enabled we just need to adjust the IP addresses and sub-net masks of the devices in the network.

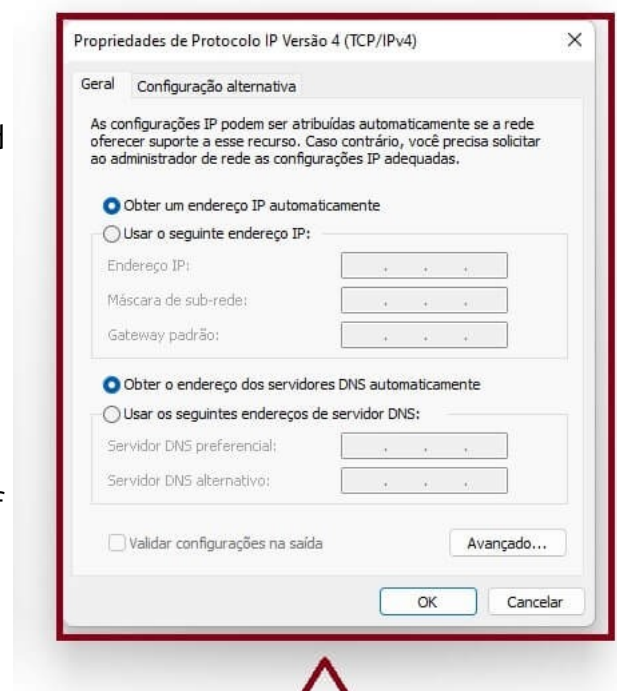


IP - its the 4 number combination where each number is [0-255] meaning it's 4 hexadecimal numbers that identify a device in a network.

sub-net mask - A sub-net mask is defined as a 32-bit address that segregates an IP address into network bits that identify the network and host bits that identify the host device operating on that network.

So since its 4 numbers meaning its 1 byte or 8 bits per number.

So the sub-net mask tells us how many of the bits represent the network and how many represent the host.



Sub-net Mask	Binary Representation	Network Bits	Host Bits
0. 0. 0. 0	0000 0000.0000 0000.0000 0000	0	32
255. 0. 0. 0	1111 1111.0000 0000.0000 0000	8	24
255.255. 0. 0	1111 1111.1111 1111.0000 0000	16	16
255.255.255. 0	1111 1111.1111 1111.1111 0000	24	8
255.255.255.255	1111 1111.1111 1111.1111 1111	32	0

Sub-net Mask	# of Usable Hosts
0. 0. 0.0	4,294,967,294
255. 0. 0.0	16,777,214
255.255. 0.0	65,534
255.255.255.0	254

Sub-net ranges and usable ips

This part is important for this project cuz much of it is resumed to connecting devices by matching sub-nets

for two devices to be connected they need to have:

1. have physical connection via cable or wireless.
2. Both need to have and IP address in the same sub-net.

So we know that the mask tell us how many bits of the IP address are used for the network but depending on that number the network can be divided into sub-networks, above we only saw full octets of 1 or 0 but we can have partially filled octets like:

1100 0000 = 192 this means that the first two bits of this octet are used for the network and the rest for the device, there are only 8 variations since these need to be **contiguous**.

00000000 → 0
10000000 → 128
11000000 → 192
11100000 → 224
11110000 → 240
11111000 → 248
11111100 → 252
11111110 → 254
11111111 → 255

The way the subdivision works is by powers of two specifically the $2^{(\text{num of bits taken})}$ so 1110 0000 has 3 bits taken so 2^3 meaning 8 and the way it works is u divide 255 into 3 intervals and the first is used for the network and the last for the broadcast so in this case we have:

$$255/8 = 31$$

[0 31] usable : [1 30]

[32 64] [33 63]

[65 97] [66 96]

[98 130] [99 129]

[131 163] [132 162]

[164 196] [165 195]

[197 229] [198 228]

[230 255] [231 254]

CIDR	Sub-net Mask	# of Sub-nets	# of Usable Hosts
/0	0.0.0.0	1	4,294,967,294
/1	128.0.0.0	2	2,147,483,646
/2	192.0.0.0	4	1,073,741,822
/3	224.0.0.0	8	536,870,910
/4	240.0.0.0	16	268,435,454
/5	248.0.0.0	32	134,217,726
/6	252.0.0.0	64	67,108,862
/7	254.0.0.0	128	33,554,430
/8	255.0.0.0	256	16,777,214
/9	255.128.0.0	512	8,388,606
/10	255.192.0.0	1,024	4,194,302
/11	255.224.0.0	2,048	2,097,150
/12	255.240.0.0	4,096	1,048,574
/13	255.248.0.0	8,192	524,286
/14	255.252.0.0	16,384	262,142
/15	255.254.0.0	32,768	131,070
/16	255.255.0.0	65,536	65,534
/17	255.255.128.0	131,072	32,766
/18	255.255.192.0	262,144	16,382
/19	255.255.224.0	524,288	8,190
...			
/30	255.255.255.252	268,435,456	2
/31	255.255.255.254	536,870,912	0
/32	255.255.255.255	4,294,967,296	1

The start point, the first IP


After we know how many host ips we have and their intervals we need to know what are is/are the actual network ips, doing this is pretty simple we just do a bit-wise and with the current IP so we can know what is the start, IP based on the mask, its essentially just looking at the network bits and seeing where they fall, for eg:

lets say we have:

..192.13 ..0000 1101

..255.252 ..1111 1100

result ..0000 1100 = 12 meaning ..192.12 is our first IP

actually an easier way to do this is just to know that the first IP is just the one where the host bits are all 0 

CIDR - Classless Inter-Domain Routing

CIDR (Classless Inter-Domain Routing) is a flexible system for specifying IP address ranges using a prefix length (e.g., `/24`), replacing rigid class-based networks (A, B, C). It's written as an IP and slash number, like `192.168.1.0/24`, where the number indicates how many bits are for the network (e.g., `/24` = 24 network bits, 8 host bits).

- **Efficiently allocates IP addresses**, avoiding waste (e.g., `/30` for 2 devices, not a whole `/24`).
- **Reduces routing table size** by allowing route aggregation (e.g., grouping /24s into a /22).
- **Offers flexibility** to create sub-nets of any power-of-2 size (4, 8, 16, etc.).
- **Maximizes IPv4 use**, crucial as addresses became scarce.
- **Simplifies network design and scaling** for modern setups like cloud or home networks.

Converting Sub-net Mask to CIDR (/X)

A sub-net mask is a 32-bit number written in dotted-decimal format e.g., 255.255.255.0).

Each 255 means all 8 bits in that octet are 1s (network portion), and each 0 means all 8 bits are 0s (host portion).

The CIDR number (/X) is simply the total number of 1 bits in the mask.

The solutions

Rule 1: diff mask cant connect!

LVL 01 - in this one we have to connect two pairs of pcs, just a simple direct connection.

a) for the lil bro pc we have the standard 255.255.255.0 on both so its pretty simple, since one IP is fixed we need to copy the network part meaning the first 3 octets and then in the last one we can assign any of the ips from 1 to 254 so long as they are different. 192.168.0.0/28

b) Here both sub-net mask are fixed and only one IP to edit, in this case since the IP takes two octets meaning that we can use any device address from

xxxx xxxx 0.0 to xxxx.xxxx.255.255 except those two specifically but we can use (0.255), (1.0) (1.255), (255.0).

LVL 02 - similar to one but a little trickier, we can get it to easy but to find all solutions its not so easy.

The fixed mask is 1110 0000 = 224, we have 5 bits for host so $2^5 = 32$ and thus 8 IP ranges starting from [0 - 31] and so on, but where do these start, for that we need to perform a bit-wise and on the values of the last octet of I the IP and sub-net mask.

Step = 32

IP 192.168.148.222 = 1101 1110

mask 255.255.255.224 = 1110 0000

result **1100 0000 = 192**

Since we are in range of 192-223 and 223 is taken, and even if it weren't its the broadcast IP, we can use any IP from 193 to 221.

For the second we have a CIDR /30 this means we have 30 bits for network meaning or minus 2 bits from the 4th octet 1111 1100 meaning we only have 4 ips per range and our start is:

step 4

IP 127. 42. 42. 42 = 0010 1010

mask 255.255.255.252 = 1111 1100

res = 0010 1000 = [40 - 43]

Ranges

[0 - 31]
[32 - 63]
[64 - 95]
[96 - 127]
[128 - 159]
[160 - 191]
[192 - 223]
[224 - 255]

Entao podemos usar apenas o 41 e o 42, if we start with this IP

LVL 3 - this one is also simple probably its just so we can be introduced to switches in this one we have one fixed mask and one fixed IP, first juts make the masks all match and then find the start using the one fixed IP and then since we have 7 bits for host that means we have 128 ips per range and only two ranges so having the fixed IP at 104.198.134.125 and a range of **[0 - 128]** we can choose any IP in this range except, 0, 128, and 125.

LVL 4 - this time we are introduced to a router, the first part is just two pcs and a switch same as before just match the masks find the range and start, since its 240 (1111 0000) that means we have 4 bits for the host $2^4 = 16$ ips per range and the start is 128 so range is **[128 - 143]**, now for the next part we have /23 and since we cant connect using diff masks just match the mask and use the same range.

LVL 5 - first need to connect A1 to R1 and B1 to R2 so just match the masks and IP ranges.

a) 128 means we have 7bits, $2^7 = 128$ ips and start is 0 so range is [0 - 127] now A connects to R1

b) 192 in the 3rd octet means we have {1100 0000} so 6 extra bits for the host plus the full 4th octet its 14 bits so $2^{14} = 16384$ ips per range and start is 128.0 up to 128.255 but this is only 255 ips, the other ips are in diff ranges in this case our range spans more variations on the third octet... [128.0] to [128.255] plus [129.0] [129.255] and in this case only 128.0 is the network the rest are usable ips, both 128.255 and 129.255, in order to know the last usable IP we divide our IP number 16384 by the intervals in this case 255 cuz [0-255] we get 64 so 128 + 64 we get 192 which is the first IP of the next range thus 191.255 is our broadcast.192.168.0.0/28

Thus our interval is [129.0] - [191.255]

c) Now for the final part we need to have both pcs communicate via the router.

So now we start using routing.

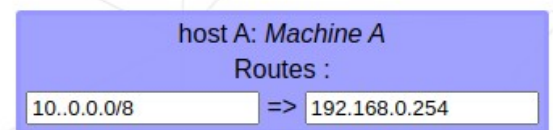
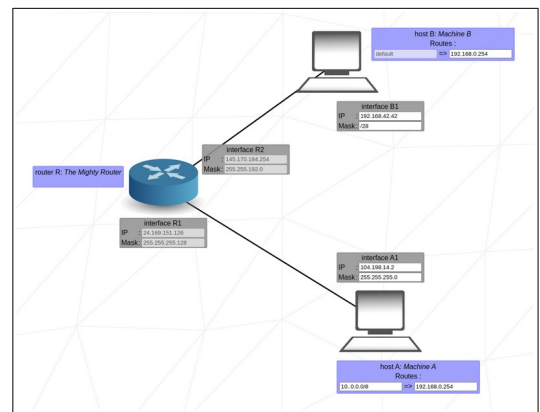
Routing

Routing is the process of selecting the best path for data packets to travel across a network from their source to their destination.

In these problems we have these new windows that say x router: y basically ur saying that packets meant for x should go to y, in case u want to say all packets u should use default = 0.0.0.0.

Also the the IP in the left side (x) must have its subnetmask CIDR following it.

In lvl 5 we just need to say for both pcs that all their traffic should be routed to the router interface that is next to them, thus A routes default => 145.170.184.254 (R2) and B routes default => 24.169.151.126 (R1) this solves the problem now A and B can communicate.



LVL 6 - Now we have the internet which acts just like a network except that the left side of the routing table we cant use /0 we need to specify the actual mask. Seems this is just something the program doesn't accept this but that in reality it should work.

OK this one got me a lil confused!

So first we connect A1 to R1, then we route all traffic from A1 to R1, so far so simple, then we have two routing tables, the one under the router I was told is to connect to another router across the internet kind of like how we see in lvl 7, that why it has that fixed IP that is nowhere in the problems its an IP that is part of some router across the web, so we just route default to it, meaning all unknown traffic .

Now for the confusing part:

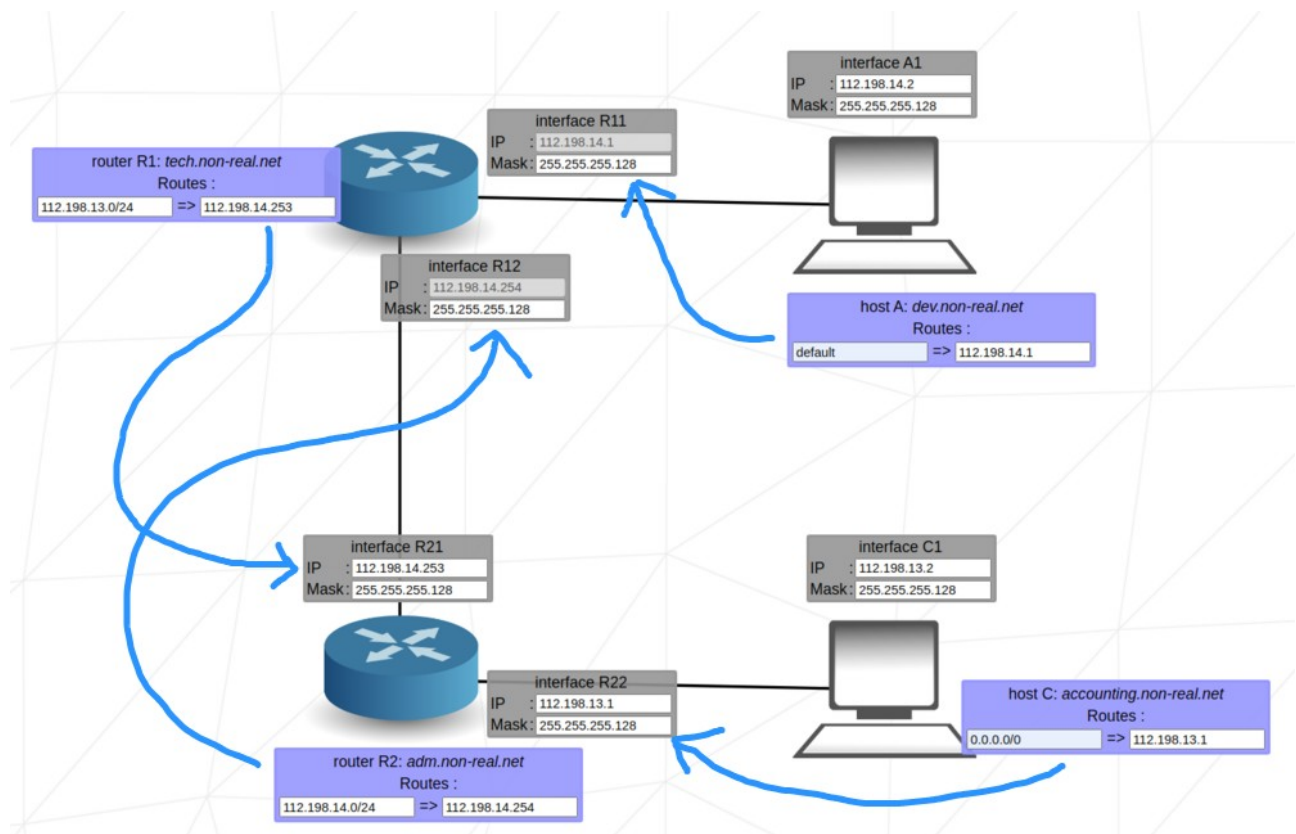
In the routing table right next to the internet symbol we are supposed to route all traffic that is going to the IP of our home network (32.14.200.0/24) to R2 and setting it this way does indeed work just set 32.14.200.0/24 => R2.

The thing is this works even If we vary the CIDR, 25 doesn't work but {0 <= x <= 24} works even though I think /0 is a special case,

why this works I have no idea, in fact /0 seems to allow any kind of x.x.x.x

LVL 7 - This case is similar to the last one, in this case we have two pcs and two router this is essentially how routing is supposed to work at its most basic, it allows us to connects across diff networks.

THIS WAS THE MOST FRUSTRATING ONE YET!! 😞😭😞



So the solution is kinda simple I just got stuck on it for a while until someone helped me out.

1 - Like in the previous ones we need to make the neighboring interfaces able to connect to one another by matching the masks and IP ranges for each section in this case we have 3.

2 - We then set the path forwarding rules, A forwards to its router's interface r1/r11 then in the router we set a forward to the neighboring router's interface r2/r21 then we do the inverse, C forwards to r2/r22 and r2 forwards to r1/r12.

Setting things up like this does work, kind of... it has a major problem, if the IP ranges of the different ips in one of the 3 networks match then we will have a kind of conflict, if our sender network matches the params set for the receiving network then we will get a

loop where the sender is receiving the package it sent over and over, in a loop.

To fix this we need to make it so the sender and receiver networks cant match, we achieve this by using a mask that will divide the IP ranges into intervals that our ips don't overlap. Actually not only should the sender and receiver networks not match but the same for the middle network the one in between the router we need to separate the 3.

So in this case we have two fixed ips one in net-A and the other in the routers

network, these are 112.198.14.1 and 112.198.14.254 that means that no matter what one is the very first

IP available for the given mask and the other the very last

now we just need to use any

number of division above 2, so a min of 3, with 3 division or ranges

we can separate our 3 nets.

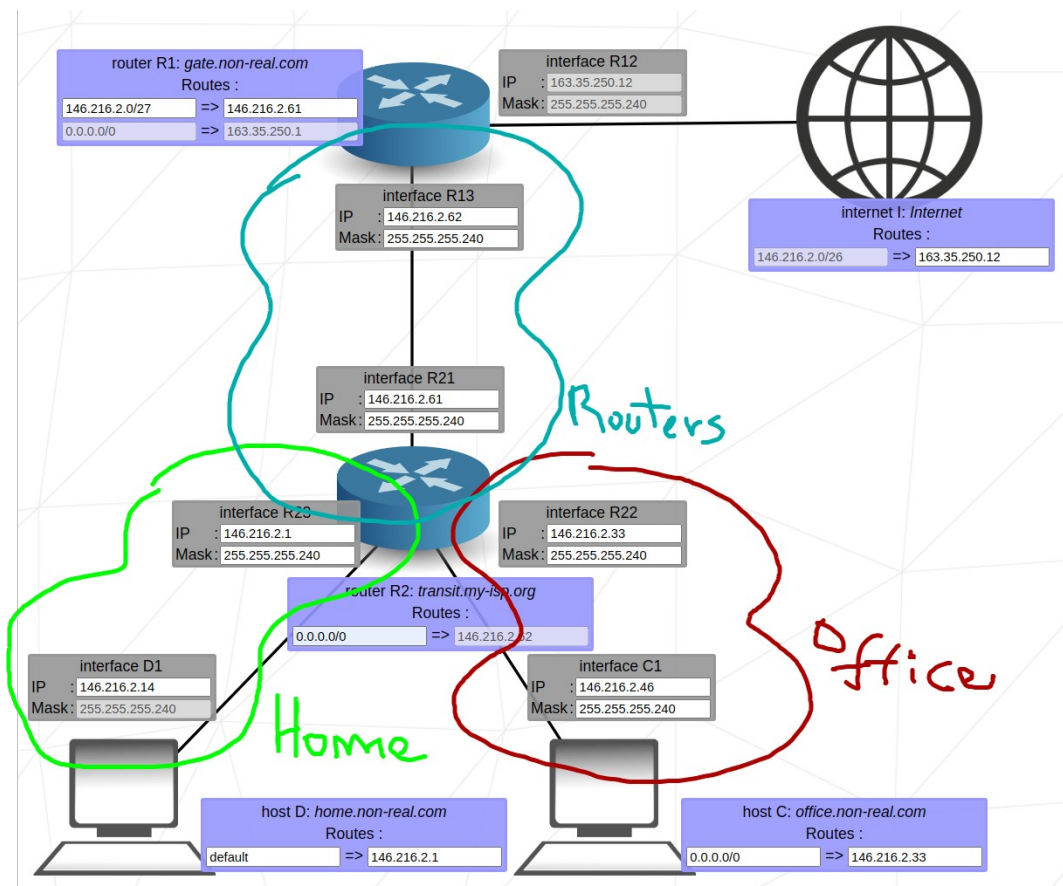
So at min our mask (use the same in all places, for simplicity)needs to be 255.255.255.192

Why this is the min mask ?

So we want to have 3 or more ranges in order to separate our 3 networks, than means we need $255 / 3 = 85$, so our number of ips per range needs to be 85 or smaller, What number of bits gets us this ? That would be $2^x = 85$ so $\log_2(85) = 6.4$ and $2^6 = 64$ so our ranges are:

[0-63]	[64-127]	[128-191]	[192-255]
[net_A]			[routers]

So now we just use one of the other ranges for net_c and we are done, we can always use smaller ranges it works as well, the important thing is to separate the ranges for each net.



LVL 8 - in this one we first get an easy one, we just need to connect two pcs in two different nets over a route.

The steps are the same as before:

1. use the same mask in all hosts in the same network. Actually in this case just go ahead and use the same mask everywhere for simplicity.
2. Use diff network parts of the IP for each network.
3. Route through the router interface in the network.

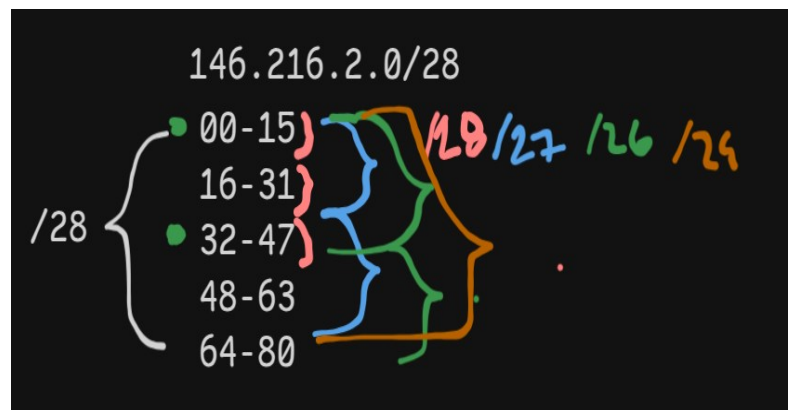
As per specific ips and masks since one of them already has a mask lets just use that for the others as well so now all pcs in both home and office sub-nets have the same mask /28 now for the ips we can use virtually any valid IP ranges given the masks. This last part is not quite true for the entire problem but try anyway it will work for objective 1.

Now for obj 2 and 3 we know that the internet is trying to reach **146.216.2.0/26** meaning this needs to be the address of our final networks (from internet to pc), thus we must change the ips of both the home and office networks to match this, but we cannot use just any ips.

We know our home and office networks must start from **146.216.2.0** and since we have the mask **255.255.255.240** we have 16 intervals of 16 ips each and we must choose one for each network:

[00 15] [16 31] [32 47]
[48 63] [64 80]

But we also need to avoid the IP range of the routers interfaces this because if they were a part of the network the internet is trying to reach then we would likely get a loop or another kind of error.



Thus we can only use the first 3 ranges for our home / office networks in order to avoid the range of the router interfaces while starting from **146.216.2.0**.

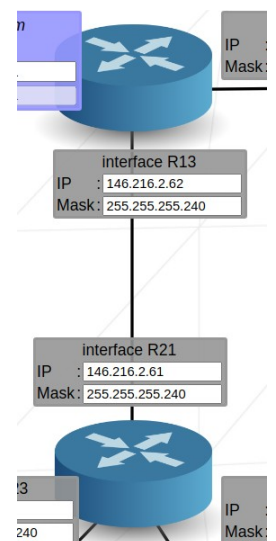
Lets use range **[00 15]** for home and **[32 47]** for office, test and obj 1 should still work, then we set the interfaces between the routers since the routing table of R2 is fixed we need to assign the its next-step (**146.216.2.62**) IP to the interface R13 of R1 and then we choose for R21 of R2 a valid IP that is in the same range, in this case **[48 63]**.

Next we set the routing table for the internet (internet I) to pass through the only interface in the same network: **163.35.250.12**.

Finally we set the R1 to route packets meant for **146.216.2.0/28** (yes use **/28** not **/26**) to the interface R21 of R2 with IP **146.216.2.49**.

So now it should say that both obj 1 and obj 3 passed but that there is no reverse way in obj 2 this is because we used **146.216.2.0/28** which means that we are looking for the receiver in the 16 ips of the range **146.216.2.[0-15]** which only matches the home network and not the office to fix this we need a mask or CIDR to encompass both the home and office networks.

Since home uses the range **[00 15]** and office **[32 47]** that means we need a range of 48 ips so $256 / 48 = 5.3 \rightarrow 6$ which means that our mask



needs 6 bits for the host which means our CIDR = $32 - 6 = 26$, this way our first range goes from [0 - 47] encompassing both [00 15] and [32 47].

With this we have the solutions the other option is to use the first two usable ranges instead of the first and the third thus we could have used for home and office [00 15] and [16 31] and since they make up a smaller range of just 32 ips and not 48 like the previous one we can use the CIDR /27.