ANSWERS

A12.5 IP addresses and subnets

Networks and the web · A level · AQA 7517 4.9.4.3 · about 30 min

What this lesson is about

Network and host identifiers, subnet masks with AND, IPv4 and IPv6, public and private addresses.

- Network and host
- Subnet masks
- IPv4 and IPv6
- Public and private addresses

Questions

6 marks in all

1. A host has the address 192.168.10.200 with subnet mask 255.255.255.224. What is its network address? [1 mark]

> **Answer:** 192.168.10.192. 200 is 11001000 and 224 is 11100000; ANDed they give 11000000, which is 192.

2. What does this program print? [1 mark]

```
address = 0b10110110
mask = 0b11110000
print(address & mask)
```

> **Answer:**
>
> ```
> 176
> ```
>
> 10110110 AND 11110000 keeps the top four bits: 10110000, which is 176.

3. How many usable host addresses does a /27 subnet have? [1 mark]

> **Answer:** 30. 32 - 27 = 5 host bits give 2^5 = 32 identifiers, minus the network and broadcast addresses.

4. Why was IPv6 introduced? [1 mark]

- ☐ **A** IPv4's 32-bit addresses were running out as more devices joined the Internet
- ☐ **B** IPv4 could not carry video
- ☐ **C** IPv4 addresses could not be written in denary
- ☐ **D** IPv4 did not support routers

> **Answer:** A. 32 bits give about 4.3 billion addresses; IPv6's 128 bits give about 3.4 x 10^38.

5. Which of these are private (non-routable) IPv4 addresses? Choose all that apply. [1 mark]

Tick every answer that is true.

- ☐ **A** 10.4.0.1
- ☐ **B** 172.20.5.5
- ☐ **C** 192.168.1.1
- ☐ **D** 172.32.0.1
- ☐ **E** 8.8.8.8

> **Answer:** A, B, C. The private blocks are 10.0.0.0/8, 172.16.0.0 to 172.31.255.255, and 192.168.0.0/16; 172.32.0.1 is outside them.

6. A host with address 10.1.1.5/24 wants to send to 10.1.2.9. What does it do? [1 mark]

- ☐ **A** Sends the packet to its default gateway, because ANDing with the mask gives a different network
- ☐ **B** Sends the packet directly, because both start with 10
- ☐ **C** Asks DHCP for a new address
- ☐ **D** Drops the packet, because 10.1.2.9 is private

> **Answer:** A. With a /24 mask the networks are 10.1.1.0 and 10.1.2.0, which differ, so the router must forward it.

The task: local or through the router?

The robot has the address `192.168.4.70` with prefix length `26`. Write `to_int(address)`, which turns a dotted IPv4 string into one integer from 0 to 2³² − 1, and `to_dotted(value)`, which turns such an integer back into a dotted string. Work the mask out from the prefix length, not by typing it. Then print, one per line: - `mask: <dotted mask>` - `network: <dotted network address>`, the robot's address AND the mask - `broadcast: <dotted broadcast address>`, the network address with every host bit set to 1 - `hosts: <n>`, the number of usable host addresses Then for each of `192.168.4.100`, `192.168.4.130`, `172.20.1.9`, `8.8.8.8` and `192.168.4.65`, in that order, print `<address> <where> <kind>`: where is `local` if it is on the robot's subnet and `router` if not; kind is `private` if it lies in one of the three private blocks and `public` otherwise. Use `&` to find network addresses. That is 9 lines in all. The robot does not drive.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def to_int(address):
    return 0

def to_dotted(value):
    return "0.0.0.0"

robot, bits = "192.168.4.70", 26
```

The hint students can ask for: Turn a dotted address into one 32-bit number and back again, so the mask, network and broadcast addresses become single bitwise operations. Two addresses are on the same subnet when ANDing each with the mask gives the same result. For private or public, check the three private blocks, using a mask for each.

A solution

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def to_int(address):
    value = 0
    for part in address.split("."):
        value = value * 256 + int(part)
    return value

def to_dotted(value):
    parts = []
    for shift in (24, 16, 8, 0):
        parts.append(str((value >> shift) & 255))
    return ".".join(parts)

def prefix_mask(bits):
    return (0xFFFFFFFF << (32 - bits)) & 0xFFFFFFFF

def is_private(address):
    n = to_int(address)
    blocks = [("10.0.0.0", 8), ("172.16.0.0", 12), ("192.168.0.0", 16)]
    for start, bits in blocks:
        if n & prefix_mask(bits) == to_int(start):
            return True
```

```
    return False

robot, bits = "192.168.4.70", 26
mask = prefix_mask(bits)
network = to_int(robot) & mask
broadcast = network | (~mask & 0xFFFFFFFF)
print("mask:", to_dotted(mask))
print("network:", to_dotted(network))
print("broadcast:", to_dotted(broadcast))
print("hosts:", 2 ** (32 - bits) - 2)
for other in ["192.168.4.100", "192.168.4.130", "172.20.1.9", "8.8.8.8", "192.168.4.65"]:
    where = "local" if to_int(other) & mask == network else "router"
    kind = "private" if is_private(other) else "public"
    print(other, where, kind)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.