Lab 2: IP Address Analyzer in Go

TA in charge: Rohit Pai

Updates

April 9, 9:41 AM: Clarified that lab 2 addresses are ipv4 only
April 14, 10:10 PM: Changed submission process for lab 2 to be Gradescope-based. Please reach out to course staff if you cannot access Gradescope.
April 16, 11:27 AM: Please submit your code only to the main branch. Also, the default branch must remain main for us to correctly clone your repo and run the autograder.

Extension form if needed:
https://docs.google.com/forms/d/e/1FAIpQLSeyIPB_UCK6MDAwQJgB7hU_Rz2AntiRFnpFIWk_RRslbrn-fQ/viewform?usp=sharing

Topics

- IP address concepts: netmasks and CIDR representation
- How to handle IP addresses in Go using the `net` package.

Tasks

1. Build a command-line Go program that:

- Accepts a CIDR block (e.g., `192.168.1.0/24`) as user input. You may assume the addresses are iPV4 only.
- Performs basic analysis of the provided network:
    - Parses and validates IP addresses and CIDR notation.
    - Calculates and displays the network address, broadcast address, and subnet mask.
    - Counts and outputs the number of available hosts on the given network.
- The network address has all of the host bits set to 0
- The broadcast address has all the host bits set to 1
- The subnet mask is a representation of the CIDR prefix, consisting of four bytes separated by dots (similar to IP addresses)

2. Program output:

- Have your program print the following to stdout (replacing the ‘x’s with the appropriate octets or other values):

```
Analyzing network: xxx.xxx.xxx.xxx/yy

Network address: xxx.xxx.xxx.xxx
Broadcast address: xxx.xxx.xxx.xxx
Subnet mask: xxx.xxx.xxx.xxx
Number of usable hosts: xxx
```

Assumption

- You may assume that the first usable IP address in a range has its host bits set to the value 1 (so if the host bits are 8 bits long, then the host bits would be 00000001). You may also assume that the highest host bits in a given range is the broadcast address.
- We will only run this utility on CIDR address ranges that are between /8 and /30 (so you can ensure that there will be at least four valid IP addresses in each range)

Example

```
$ go run ipanalyzer.go 192.168.1.0/24
Analyzing network: 192.168.1.0/24

Network address: 192.168.1.0
Broadcast address: 192.168.1.255
Subnet mask: 255.255.255.0
Number of usable hosts: 254
```

2. Determine if a provided IP address is within a specified subnet. Here the first argument is a CIDR address range, and the second argument is an IP address. You should return true if the provided IP address is in the given range, and false if not.

```
$ go run ipanalyzer.go 192.168.1.0/24 192.168.1.16
true
```

```
$ go run ipanalyzer.go 192.168.1.0/24 10.0.1.2
false
```

Guidance and Hints

- Some commands in Go's 'net' package that may be useful:
    - `net.ParseCIDR()`
    - `net.IPNet`
    - `net.IP.Mask()`

Testing

The test cases cover a range of simple scenarios involving CIDR blocks from /8 to /30. Each test checks whether the correct analysis is performed on the CIDR block and whether the correct output is produced. For IP address checks, both positive and negative cases are included — verifying situations where the IP address is within the CIDR range, as well as cases where it is correctly identified as outside the range.

Submission

~~In addition to submitting your code, please include a file called report.pdf in the main directory of your submission that contains the answers to the following questions:~~

After some discussion with students and the course staff we will be changing the submission process. We will be using Gradescope and GitHub together to facilitate grading.

In addition to making sure your repo is up-to-date and part of the classroom, please also add the repo link to the Gradescope assignment "Lab 2 Repo". Then answer the questions below and copy your answers to "Lab 2 Report" (this means you no longer need to have report.pdf in your repo itself, though this is still recommended as a form of documentation).

Q1. Is 192.168.129.51 inside the address range 192.168.128.1/22 ?

Q2. How many addresses could be assigned to servers/hosts in the subnet 192.168.128.1/22 ?

Q3. You have been assigned the IP block 137.110.222.0/24. Your start-up requires subnets for four separate internal networks, each supporting at least 48 hosts. **Q3.1**: Propose a subnetting scheme that satisfies these requirements. **Q3.2**: What is the new subnet mask? **Q3.3**: Clearly identify each subnet's network address, broadcast address, and range of usable IP addresses.

Starter code

- To begin the project, please accept the GitHub invitation and clone your repository. ~~Don't forget to commit and push your code and lab report to GitHub when you're done. Doing that "submits" your codE~~ (see above)

Expected effort level

To set expectations for how much work is involved, we can say that our reference solution of ipanalyzer.go was about 70 lines of code.

###