Clark Schwarz

Code Documentation for Ping Program

Overview:

This program sends ICMP Echo Request packets (pings) to a specified server. The program is configurable via command-line arguments for controlling packet size, wait time, the number of packets to send, and timeout duration. It will print the results of the pings to the console.

The program has two primary functions:

1. **`ping`**: Handles the core functionality of sending ping requests and displaying results.
2. **`main`**: Manages command-line argument parsing and passes the parameters to the `ping` function.

`ping` Function

Description:
Sends ICMP Echo Request (ping) packets to the specified destination server.

Parameters:

- `destination (str)`: The destination server's IP address or domain to ping.
- `packets_to_receive (int)`: The number of successful pings to receive before quitting. If set to 0, the program will run indefinitely until the timeout.
- `wait_time (int)`: The number of seconds to wait between each ping request.
- `data_to_send (int)`: The size (in bytes) of the data payload in each ping packet.
- `timeout (int)`: The number of seconds the program will run before exiting, regardless of the number of successful pings.

Functionality:

- Continuously sends ping requests to the destination server until one of the following conditions is met:
    - The specified number of successful packets is received.
    - The specified timeout duration has been reached.

`main` Function

Description:
Parses command-line arguments and passes them to the `ping` function to initiate the pinging process.

Command-line arguments:

- `-c` (int): Number of echo response packets to receive before quitting (default: infinite if not provided).
- `-i` (int): The number of seconds to wait between each ping request (default: 1 second if not provided).
- `-s` (int): The size of the data (in bytes) to send with each ping request (default: 56 bytes if not provided).
- `-t` (int): The timeout duration in seconds before the program automatically exits (default: infinite if not provided).
- `-d` (str): The server (IP address or domain) to ping.

Functionality:

- The function parses the arguments and passes them to the `ping` function to perform the desired ping operations.

Example of Usage:

bash
Copy
```
python ping_program.py -d google.com -c 4 -i 2 -s 64 -t 10
```

This command will send 4 pings to `google.com` with:

- A 2-second interval between pings.
- A data payload of 64 bytes for each ping.
- A timeout of 10 seconds, after which the program will exit.

Code Documentation for Traceroute Program

Overview:

This program traces the route to a specified destination by sending ICMP Echo Request packets with increasing time-to-live (TTL) values. It reports the hops (routers) the packet passes

through to reach the destination. The program also allows customization via command-line arguments, such as controlling the number of probes per TTL and whether to display unanswered probes.

The program has two primary functions:

1. **`trace_route`**: Handles the core functionality of tracing the route and displaying the hops.
2. **`main`**: Manages command-line argument parsing and passes the parameters to the `trace_route` function.

`trace_route` Function

Description:
Traces the route to the specified destination by sending ICMP Echo Request packets with increasing TTL values. The function prints out the hops (routers) visited by the packet. Optionally, it can also display probes that were not answered based on the user's input.

Parameters:

- `destination (str)`: The destination IP address or domain to trace the route to.
- `hop_toggle (int)`: If `0`, the hop addresses are displayed symbolically and numerically; if `1`, only numerical addresses are shown.
- `probe_number (int)`: The number of probes to send per TTL value.
- `summary (int)`: If `1`, the function will print a summary of unanswered probes at the end.

Functionality:

- The function sends ICMP Echo Request packets with increasing TTL values.
- It tracks the hops the packet takes and prints the address of each hop.
- If a hop does not respond within the timeout period, it is recorded as "Timed out."
- The number of probes per TTL can be customized, and the function stops after the specified number of probes.
- Optionally, it can display a summary of hops that did not respond.

`main` Function

Description:
Parses command-line arguments and passes them to the `trace_route` function to initiate the traceroute operation.

Command-line arguments:

- `-n` (int): A flag that indicates whether the hop addresses should be printed only numerically (0) or both symbolically and numerically (1).
- `-q` (int): Sets the number of probes to send per TTL (time-to-live). The default is 1.
- `-s` (int): Controls the summary output. If `1`, prints the number of unanswered probes. If `0`, hides the summary.
- `-d` (str): The destination to trace the route to, specified as an IP address or domain name.

Functionality:

- The function parses the command-line arguments and passes them to the `trace_route` function.
- It ensures default values are used if any arguments are not provided.

Example of Usage:

bash
Copy

```
python traceroute_program.py -d google.com -n 1 -q 3 -s 1
```

This command will trace the route to `google.com` with:

- Only numerical hop addresses displayed (`-n 1`).
- 3 probes per TTL (`-q 3`).
- A summary of unanswered probes will be shown at the end (`-s 1`).