Ping and Traceroute

Cedric Bone

Feb 28, 2025

CONTENTS:

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

CHAPTER

ONE

MY_PING MODULE

my_ping.py

Python implementation of ping command that tests network connectivity. Sends ICMP echo requests and measures round-trip time.

Usage:

python my_ping.py [-c COUNT] [-i WAIT] [-s PACKETSIZE] [-t TIMEOUT] destination

my_ping.**handle_interrupt**(*signum*, *frame*, *sent_count*, *received_count*, *rtts*, *destination*)

Handle Ctrl+C interrupt. Print statistics and exit gracefully.

Parameters:

signum: Signal number frame: Current stack frame sent_count: Total packets sent received_count: Total responses received rtts: List of round-trip times destination: Target hostname or IP

my_ping.**main**()

Parse command line arguments and execute ping. Sends ICMP echo requests and displays results.

my_ping.**print_stats**(*sent_count*, *received_count*, *rtts*, *destination*)

Print ping statistics summary.

Parameters:

sent_count: Total number of packets sent received_count: Total number of responses received rtts: List of round-trip times in milliseconds destination: Target hostname or IP

my_ping.**send_ping**(*dest_ip*, *timeout*, *packet_size*, *sent_count*, *received_count*)

Send a single ping and wait for the response.

Parameters:

dest_ip: Destination IP address timeout: Maximum wait time for response packet_size: Size of ICMP payload in bytes sent_count: Number of packets sent so far received_count: Number of responses received so far

Returns:

tuple: (success_bool, rtt_ms)

CHAPTER

TWO

MY_TRACEROUTE MODULE

my_traceroute.py

Python implementation of traceroute command that maps network path to destination. Sends UDP probes with incrementing TTL values to discover intermediate routers.

Usage:

sudo python my_traceroute.py [-n] [-q NQUERIES] [-S] [-f FIRST_TTL] [-m MAX_TTL] [-w WAIT] destination

my_traceroute.**main**()

Parse command line arguments and execute traceroute. Shows network path to destination with response times for each hop.

my_traceroute.**send_probe**(*send_socket*, *recv_socket*, *dest_ip*, *ttl*, *port*, *timeout*)

Send a single traceroute probe and wait for response.

Parameters:

send_socket: Socket for sending UDP probes recv_socket: Socket for receiving ICMP responses dest_ip: Destination IP address ttl: Time-to-live value for this probe port: UDP destination port timeout: Maximum wait time for response

Returns:

tuple: (responding_ip, elapsed_time_ms) or (None, None) on timeout

PYTHON MODULE INDEX

m

INDEX