ping_and_traceroute

Ethan Iannicelli

Feb 12, 2025

CONTENTS:

CHAPTER

ONE

PING SUMMARY

To get the checksum of an ICMP packet based on the string representation of the packet, use the `checksum()` function:

my_ping.**checksum**(*data*)

Creates the checksum for a given data for icmp packet

Parameters
data (*String*) – the input data for the checksum

Returns
calculated checksum

Return type
bitstring

To create a ping icmp packet based on a packet id and size, use the `create_packet()` function:

my_ping.**create_packet**(*id*, *size*)

Create a packet with a given id and of a given size

Parameters

- **id** (*string*) – id of the new packet
- **size** (*int*) – size of the new packet

Returns
the new icmp packet

Return type
network packet

To send a ping to a target ip address, use the `send_ping()` function:

my_ping.**send_ping**(*target*, *packetsize*)

Send a recieve a packet to a given target (of a given packetsize)

Parameters

- **target** (*string*) – the target destination
- **packetsize** (*int*) – size of packets to be used as the pings

Returns
status of this ping

Return type
boolean

To recieve a ping echo response, use the `receive_ping()` function:

my_ping.**receive_ping**(*sock*, *packet_id*, *packetsize*, *timeout=10*)

recieve a icmp ping echo response. the socket and packet_id are provided, so we know what to look for. A default timeout of 10 seconds is also applied, which should be plenty for any address that is known to be online

Parameters

- **sock** (*socket*) – the socket that is prepared to accept the echo response
- **packet_id** (*string*) – id of the incoming echo response packet

Returns
a tuple of the target address, rtt, and size of icmp packet

Return type
tuple(string, double, int)

To initialize the parser for a ping program, use the `initialize_parser()` function:

my_ping.**initialize_parser**()

initialize the parser for this program. The only required argument is the 'target' which is the target ip address to be pinged. Optional arguments include count, wait, packetsize, and timeout

Returns
fully initialized parser

Return type
parser

To handle a timeout event, use the `timeout_handler()` function:

my_ping.**timeout_handler**(*signum*, *frame*)

handler for a program timeout. calls os._exit() to avoid raising an error, as this can be called as part of an expected functionality