Ping and Traceroute Docs

version

Kirti Sharma

February 17, 2025

Contents

Welcome to Ping and traceroute's documentation!

ping module

Python implementation of the ping utility.

This module provides functionality to ping a specified host.

`ping.`**`ping`**(`host`, `count`, `interval`, `packet_size`, `timeout`)

Ping a specified host.
This function sends ICMP Echo Request packets to a specified host and reports on the round-trip time and packet loss statistics.

Parameters:

- **host** (*str*) – Hostname or IP address to ping
- **count** (*int*) – Number of ping requests to send (0 for infinite)
- **interval** (*float*) – Time interval between ping requests in seconds
- **packet_size** (*int*) – Size of the ping packet in bytes
- **timeout** (*float*) – Timeout for each ping request in seconds

traceroute module

Python implementation of the traceroute utility.

This module provides functionality to trace the route to a specified host, showing the path that packets take to reach the destination.

`traceroute.`**`traceroute`**(`dest_addr`, `max_hops=30`, `timeout=1`, `queries=3`, `numeric=False`, `summary=False`)

Perform a traceroute to a specified destination.
This function sends packets with increasing TTL values to discover the path to the destination and measure round-trip times for each hop.

Parameters:

- **dest_addr** (*str*) – Destination hostname or IP address
- **max_hops** (*int*) – Maximum number of hops to probe (default: 30)
- **timeout** (*float*) – Timeout for each probe in seconds (default: 1)
- **queries** (*int*) – Number of queries per hop (default: 3)
- **numeric** (*bool*) – If True, print numeric addresses only (default: False)
- **summary** (*bool*) – If True, print summary of unanswered probes (default: False)

utils module

Utility functions for network operations.

This module provides common functions used in ping and traceroute implementations, including packet creation, checksum calculation, and DNS resolution.

`utils.`**`calculate_checksum`**(`data`)

Calculate the checksum for an ICMP packet.

Parameters: **data** (*bytes*) – Data to calculate checksum for
Returns: Calculated checksum
Return type: int

`utils.`**`create_packet`**(`id`, `seq`, `payload_size`)

Create an ICMP Echo Request packet.

Parameters:
- **id** (*int*) – Identifier for the packet
- **seq** (*int*) – Sequence number for the packet
- **payload_size** (*int*) – Size of the packet payload

Returns: Bytes object representing the ICMP packet

Return type: bytes

`utils.`**`get_hostname`**`(ip_address)`
Get the hostname for a given IP address.

Parameters: **ip_address** (*str*) – IP address to lookup

Returns: Hostname if found, otherwise the original IP address

Return type: str

`utils.`**`resolve_hostname`**`(hostname)`
Resolve a hostname to its IP address.

Parameters: **hostname** (*str*) – Hostname to resolve

Returns: IP address of the hostname, or None if unresolvable

Return type: str or None

`utils.`**`setup_socket`**`()`
Set up a raw socket for ICMP communication.

Returns: Configured socket object

Return type: socket.socket

Indices and tables

- **genindex**
- **modindex**
- **search**

Index

Python Module Index

p

ping

t

traceroute

u

utils