Traceroute Lab

Ze He(zh700@nyu.edu)



Code:

|  |
| --- |
| from socket import \*  import socket  import os  import sys  import struct  import time  import select  import binascii  ICMP\_ECHO\_REQUEST = 8  MAX\_HOPS = 30  TIMEOUT = 2.0  TRIES = 2  #ID = 1  def checksum(str):  csum = 0  countTo = (len(str) / 2) \* 2    count = 0  while count < countTo:  thisVal = ord(str[count+1]) \* 256 + ord(str[count])  csum = csum + thisVal  csum = csum & 0xffffffffL  count = count + 2    if countTo < len(str):  csum = csum + ord(str[len(str) - 1])  csum = csum & 0xffffffffL    csum = (csum >> 16) + (csum & 0xffff)  csum = csum + (csum >> 16)  answer = ~csum  answer = answer & 0xffff  answer = answer >> 8 | (answer << 8 & 0xff00)  return answer  def build\_packet():  myChecksum = 0  #global ID  ID = os.getpid() & 0xFFFF  header = struct.pack("bbHHh", ICMP\_ECHO\_REQUEST, 0, myChecksum, 0, 1)  #ID += 1  data = struct.pack("d", time.time())  myChecksum = checksum(header + data)  if sys.platform == 'darwin':  myChecksum = socket.htons(myChecksum) & 0xffff  else:  myChecksum = socket.htons(myChecksum)  header = struct.pack("bbHHh", ICMP\_ECHO\_REQUEST, 0, myChecksum, ID, 1)  return header + data    def get\_route(hostname):  timeLeft = TIMEOUT  for ttl in xrange(1,MAX\_HOPS):  for tries in xrange(TRIES):  destAddr = gethostbyname(hostname)  #fill in start  #make a raw socket named mySocket  myicmp = socket.getprotobyname("icmp")  mySocket = socket.socket(socket.AF\_INET, socket.SOCK\_RAW, myicmp)  #fill in end  mySocket.setsockopt(IPPROTO\_IP, IP\_TTL, struct.pack('I', ttl))  mySocket.settimeout(TIMEOUT)  try:  d = build\_packet()  mySocket.sendto(d, (hostname, 0))  t = time.time()  startedSelect = time.time()  whatReady = select.select([mySocket], [], [], timeLeft)  howLongInSelect = (time.time() - startedSelect)  if whatReady[0] == []:  print" \* \* \* Request timed out"  recvPacket, addr = mySocket.recvfrom(1024)  timeReceived = time.time()  timeLeft = timeLeft - howLongInSelect  if timeLeft <= 0:  print "\* \* \* Request timed out."  except timeout:  continue  else:  # fetch the icmp type from the IP packet  #fill in start  icmpHeader = recvPacket[20:28]  type, code, checksum, icmpid, sequence = struct.unpack("bbHHh", icmpHeader)  #fill in end  if type == 11:  bytes = struct.calcsize("d")  timeSent = struct.unpack("d", recvPacket[28:28 + bytes])[0]  print " %d rtt = %.0f ms %s" %(ttl,(timeReceived - t) \*1000 , addr[0])  elif type == 3:  bytes = struct.calcsize("d")  timeSent = struct.unpack("d", recvPacket[28:28 + bytes])[0]  print " %d rtt = %.0f ms %s" %(ttl,(timeReceived - t) \*1000 , addr[0])  elif type == 0:  bytes = struct.calcsize("d")  timeSent = struct.unpack("d", recvPacket[28:28 + bytes])[0]  print " %d rtt = %.0f ms %s" %(ttl,(timeReceived - t) \*1000 , addr[0])  return  else:  print "error"  break  finally:  mySocket.close()  print("www.google.com")  get\_route("www.google.com")  print("www.baidu.com")  get\_route("www.baidu.com")  print("www.bbc.com")  get\_route("www.bbc.com")  print("www.australia.com")  get\_route("www.australia.com") |