# Chapter 8:
# Traceroute Program

## Introduction

- ❑ *traceroute* **program: by Van Jacobson**
- ❑ **Is a handy debugging tool that allows us to further explore the TCP/IP protocol.**
- ❑ **Features:**
  - ❖ To see the route that IP datagrams follow from one host to another
  - ❖ To let us use the IP source route option.

## traceroute **Program Operation**

- **Why NOT just extend Ping program in IP record route option:**
  - Not all routers have supported the record route option
  - Record Route is normally one-way option. Most implementations of the Ping server reflect an incoming RR list, but this doubles the number of IP addresses recorded.
  - The room (9 IP addresses in the IP header) allocated for options in the IP header isn't large enough today to handle most routes

## traceroute **Program Operation (Cont.)**

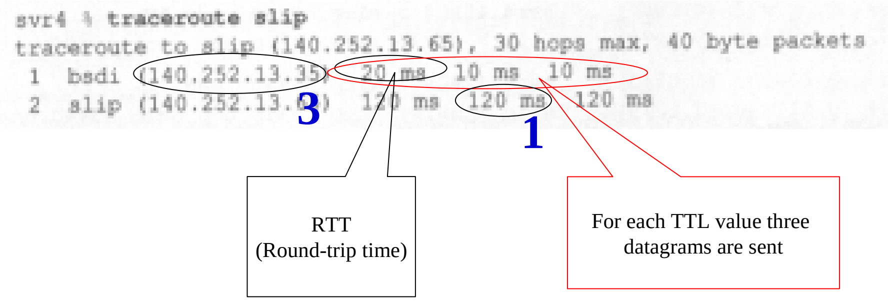
- traceroute **uses ICMP and the TTL field (generally, initial value: 64) in the IP header.**
- **The TTL field has effectively become a hop counter, decremented by one by each router.**
- **The purpose of the TTL field is to prevent datagram from ending up in infinite loops, which can occur during routing transients.**

## traceroute **Program Operation (Cont.)**

- traceroute **principles:**
  - use TTL field:
    - a router gets a IP datagram whose TTL is either 0 or 1 => not forward it and throws away AND send back to the originating host an ICMP “*time exceeded*”
  - use UDP:
    - assign an unlikely value (>30000) to the port number AND even if the datagram REALLY reached the destination, it also caused a ICMP “*port unreachable*”
  - operations:
    - 1. Set TTL=1, send the IP datagram and then gets a ICMP from the FIRST router
    - 2. Set TTL=2, and then gets the address of the second router
    - 3. And so on for TTL=N, but if the error is “*port unreachable*” then we know reached the destination

## LAN Output

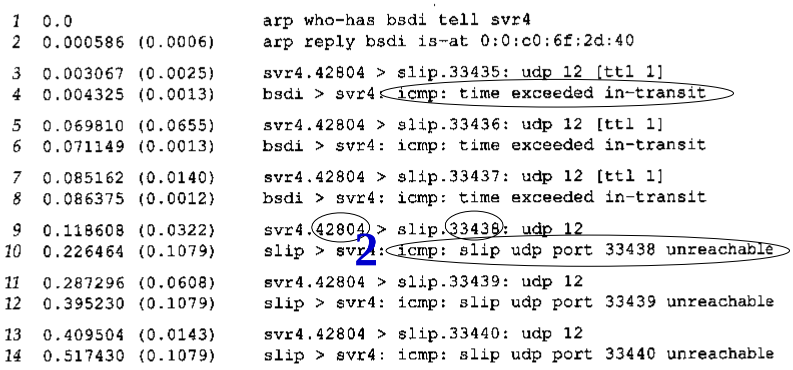
- **LAN example:**



```
svr4 % traceroute slip
traceroute to slip (140.252.13.65), 30 hops max, 40 byte packets
 1  bsdi (140.252.13.35)  20 ms  10 ms  10 ms
 2  slip (140.252.13.65)  120 ms  120 ms  120 ms
```

## LAN Output (Cont.)

- **Tcpdump:**



```
 1  0.0                 arp who-has bsdi tell svr4
 2  0.000586 (0.0006)   arp reply bsdi is-at 0:0:c0:6f:2d:40

 3  0.003067 (0.0025)   svr4.42804 > slip.33435: udp 12 [ttl 1]
 4  0.004325 (0.0013)   bsdi > svr4: icmp: time exceeded in-transit

 5  0.069810 (0.0655)   svr4.42804 > slip.33436: udp 12 [ttl 1]
 6  0.071149 (0.0013)   bsdi > svr4: icmp: time exceeded in-transit

 7  0.085162 (0.0140)   svr4.42804 > slip.33437: udp 12 [ttl 1]
 8  0.086375 (0.0012)   bsdi > svr4: icmp: time exceeded in-transit

 9  0.118608 (0.0322)   svr4.42804 > slip.33438: udp 12
10  0.226464 (0.1079)   slip > svr4: icmp: slip udp port 33438 unreachable

11  0.287296 (0.0608)   svr4.42804 > slip.33439: udp 12
12  0.395230 (0.1079)   slip > svr4: icmp: slip udp port 33439 unreachable

13  0.409504 (0.0143)   svr4.42804 > slip.33440: udp 12
14  0.517430 (0.1079)   slip > svr4: icmp: slip udp port 33440 unreachable
```

2

**Figure 8.1** tcpdump output for traceroute example from svr4 to slip.

## LAN  Output (Cont.)

- **1: the calculation of the RTT should be for the SLIP link:**
  - SLIP link speed = 960 bytes/sec
  - the size a sent UDP datagram = 42 bytes
    - 12 bytes (Data, sequence number+a copy of the outgoing TTL+ the time at which the datagram was sent)
    - 20 bytes (IP header)
    - 8 bytes (UDP header)
    - 2 bytes (at least, of SLIP framing)

## LAN Output (Cont.)

- **The size of a sent back ICMP datagram = 58 bytes**
  - 20 bytes (IP header)
  - 8 bytes (ICMP message)
  - 20 + 8 bytes (the IP header of the error datagram and the first 8 bytes of data of the error part after IP header)
  - 2 bytes (at least, of SLIP framing)
- **Expected RTT = (42+58)/960 =~ 104 ms**

## LAN Output (Cont.)

- **2: the source port number (42804) seems high:**
  - Because the source port number = pid | 32768 (logical OR)
- **3: the source IP address in the returned ICMP message is the IP address of the interface:**

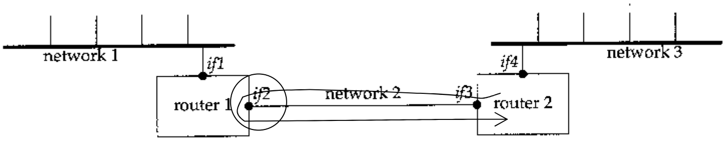
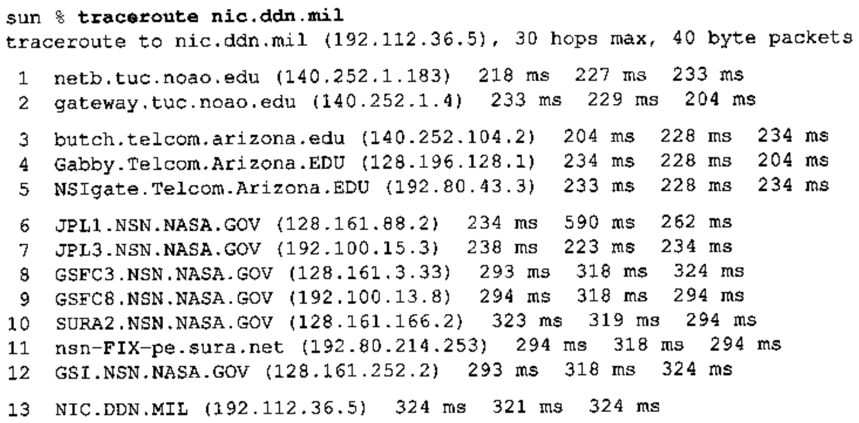


**Figure 8.3** Identification of interfaces printed by `traceroute`.

## WAN Output

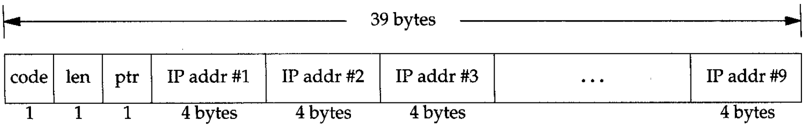


```
sun % traceroute nic.ddn.mil
traceroute to nic.ddn.mil (192.112.36.5), 30 hops max, 40 byte packets

 1  netb.tuc.noao.edu (140.252.1.183)  218 ms  227 ms  233 ms
 2  gateway.tuc.noao.edu (140.252.1.4)  233 ms  229 ms  204 ms

 3  butch.telcom.arizona.edu (140.252.104.2)  204 ms  228 ms  234 ms
 4  Gabby.Telcom.Arizona.EDU (128.196.128.1)  234 ms  228 ms  204 ms
 5  NSIgate.Telcom.Arizona.EDU (192.80.43.3)  233 ms  228 ms  234 ms

 6  JPL1.NSN.NASA.GOV (128.161.88.2)  234 ms  590 ms  262 ms
 7  JPL3.NSN.NASA.GOV (192.100.15.3)  238 ms  223 ms  234 ms
 8  GSFC3.NSN.NASA.GOV (128.161.3.33)  293 ms  318 ms  324 ms
 9  GSFC8.NSN.NASA.GOV (192.100.13.8)  294 ms  318 ms  294 ms
10  SURA2.NSN.NASA.GOV (128.161.166.2)  323 ms  319 ms  294 ms
11  nsn-FIX-pe.sura.net (192.80.214.253)  294 ms  318 ms  294 ms
12  GSI.NSN.NASA.GOV (128.161.252.2)  293 ms  318 ms  324 ms

13  NIC.DDN.MIL (192.112.36.5)  324 ms  321 ms  324 ms
```

**Figure 8.4** traceroute from host sun to nic.ddn.mil.

## IP Source Routing Option

- **Source routing: the sender specifies the route:**
  - Strict: the sender specifies the exact path that the IP datagram must follow. If a router encounters a next hop in the source route that isn't on a directly connected network, an ICMP "*source route failed*" error is returned.
  - Loose: the sender specifies a list of IP address that the datagram must traverse, but the datagram can also pass through other routers between any two addresses in the list



39 bytes

| code | len | ptr | IP addr #1 | IP addr #2 | IP addr #3 | . . . | IP addr #9 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 4 bytes | 4 bytes | 4 bytes | | 4 bytes |

**Figure 8.6** General format of the source route option in the IP header.

# IP Source Routing Option (Cont.)

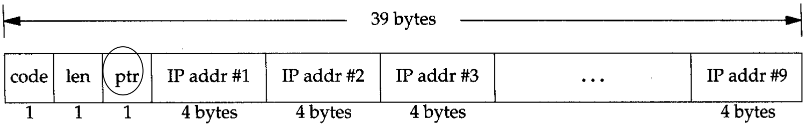


**Figure 8.6** General format of the source route option in the IP header.

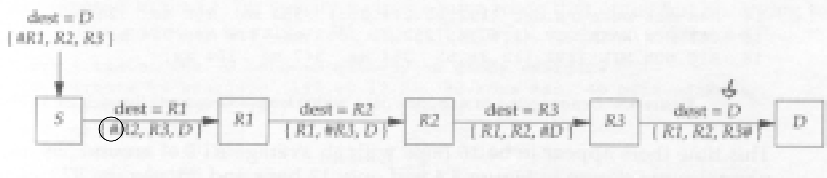
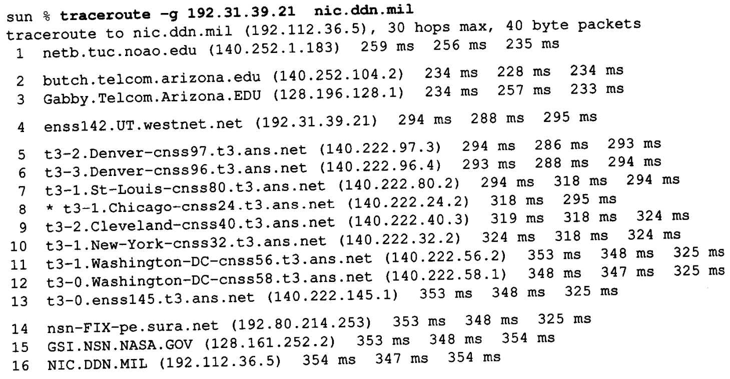


**Figure 8.7** Example of IP source routing.

# Traceroute Examples with Loose Source Routing

- -g option: loose source routing

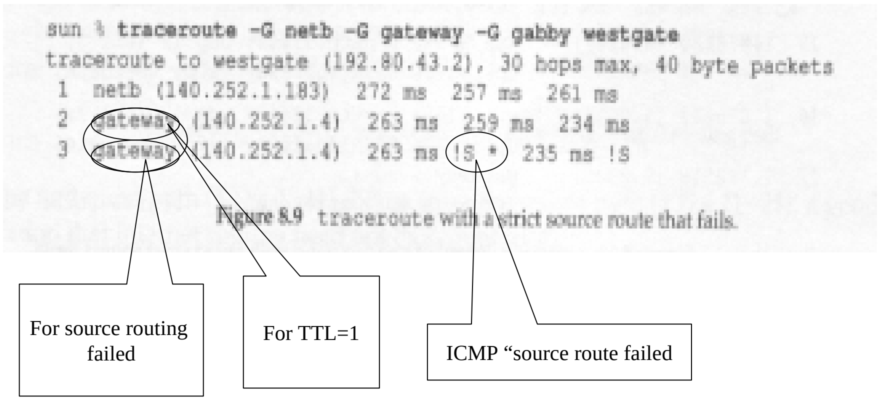


```
sun % traceroute -g 192.31.39.21  nic.ddn.mil
traceroute to nic.ddn.mil (192.112.36.5), 30 hops max, 40 byte packets
 1  netb.tuc.noao.edu (140.252.1.183)  259 ms  256 ms  235 ms

 2  butch.telcom.arizona.edu (140.252.104.2)  234 ms  228 ms  234 ms
 3  Gabby.Telcom.Arizona.EDU (128.196.128.1)  234 ms  257 ms  233 ms

 4  enss142.UT.westnet.net (192.31.39.21)  294 ms  288 ms  295 ms

 5  t3-2.Denver-cnss97.t3.ans.net (140.222.97.3)  294 ms  286 ms  293 ms
 6  t3-3.Denver-cnss96.t3.ans.net (140.222.96.4)  293 ms  288 ms  294 ms
 7  t3-1.St-Louis-cnss80.t3.ans.net (140.222.80.2)  294 ms  318 ms  294 ms
 8  * t3-1.Chicago-cnss24.t3.ans.net (140.222.24.2)  318 ms  295 ms
 9  t3-2.Cleveland-cnss40.t3.ans.net (140.222.40.3)  319 ms  318 ms  324 ms
10  t3-1.New-York-cnss32.t3.ans.net (140.222.32.2)  324 ms  318 ms  324 ms
11  t3-1.Washington-DC-cnss56.t3.ans.net (140.222.56.2)  353 ms  348 ms  325 ms
12  t3-0.Washington-DC-cnss58.t3.ans.net (140.222.58.1)  348 ms  347 ms  325 ms
13  t3-0.enss145.t3.ans.net (140.222.145.1)  353 ms  348 ms  325 ms

14  nsn-FIX-pe.sura.net (192.80.214.253)  353 ms  348 ms  325 ms
15  GSI.NSN.NASA.GOV (128.161.252.2)  353 ms  348 ms  354 ms
16  NIC.DDN.MIL (192.112.36.5)  354 ms  347 ms  354 ms
```

**Figure 8.8** traceroute to nic.ddn.mil with a loose source route through the NSFNET.

## Traceroute Example with Strict Source Routing

- -G option: strict source routing



```
sun % traceroute -G netb -G gateway -G gabby westgate
traceroute to westgate (192.80.43.2), 30 hops max, 40 byte packets
 1  netb (140.252.1.183)  272 ms  257 ms  261 ms
 2  gateway (140.252.1.4)  263 ms  259 ms  234 ms
 3  gateway (140.252.1.4)  263 ms !S *  235 ms !S
```

Figure 8.9 traceroute with a strict source route that fails.

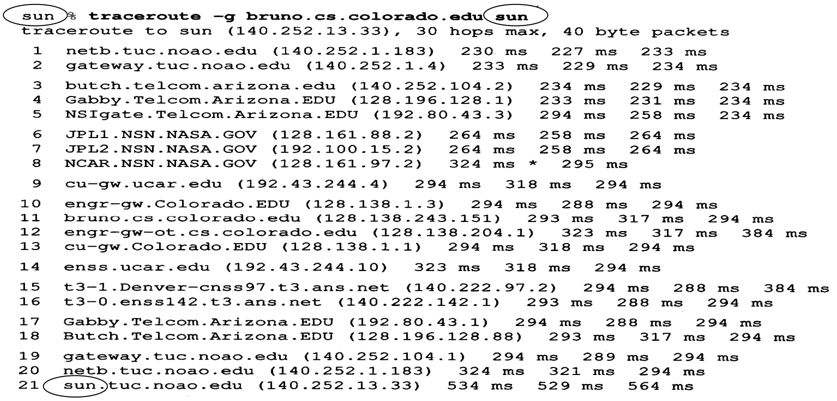
For source routing failed

For TTL=1

ICMP “source route failed

## Traceroute Round Trips with Loose Source Routing

- Routing need not be symmetrical:



```
sun % traceroute -g bruno.cs.colorado.edu sun
traceroute to sun (140.252.13.33), 30 hops max, 40 byte packets
 1  netb.tuc.noao.edu (140.252.1.183)  230 ms  227 ms  233 ms
 2  gateway.tuc.noao.edu (140.252.1.4)  233 ms  229 ms  234 ms
 3  butch.telcom.arizona.edu (140.252.104.2)  234 ms  229 ms  234 ms
 4  Gabby.Telcom.Arizona.EDU (128.196.128.1)  233 ms  231 ms  234 ms
 5  NSIgate.Telcom.Arizona.EDU (192.80.43.3)  294 ms  258 ms  234 ms
 6  JPL1.NSN.NASA.GOV (128.161.88.2)  264 ms  258 ms  264 ms
 7  JPL2.NSN.NASA.GOV (192.100.15.2)  264 ms  258 ms  264 ms
 8  NCAR.NSN.NASA.GOV (128.161.97.2)  324 ms *  295 ms
 9  cu-gw.ucar.edu (192.43.244.4)  294 ms  318 ms  294 ms
10  engr-gw.Colorado.EDU (128.138.1.3)  294 ms  288 ms  294 ms
11  bruno.cs.colorado.edu (128.138.243.151)  293 ms  317 ms  294 ms
12  engr-gw-ot.cs.colorado.edu (128.138.204.1)  323 ms  317 ms  384 ms
13  cu-gw.Colorado.EDU (128.138.1.1)  294 ms  318 ms  294 ms
14  enss.ucar.edu (192.43.244.10)  323 ms  318 ms  294 ms
15  t3-1.Denver-cnss97.t3.ans.net (140.222.97.2)  294 ms  288 ms  384 ms
16  t3-0.enss142.t3.ans.net (140.222.142.1)  293 ms  288 ms  294 ms
17  Gabby.Telcom.Arizona.EDU (192.80.43.1)  294 ms  288 ms  294 ms
18  Butch.Telcom.Arizona.EDU (128.196.128.88)  293 ms  317 ms  294 ms
19  gateway.tuc.noao.edu (140.252.104.1)  294 ms  289 ms  294 ms
20  netb.tuc.noao.edu (140.252.1.183)  324 ms  321 ms  294 ms
21  sun.tuc.noao.edu (140.252.13.33)  534 ms  529 ms  564 ms
```

**Figure 8.11** traceroute example showing unsymmetrical routing path.

## Summary

- ❑ **Traceroute:**
  - ❖ features
  - ❖ principles
  - ❖ source routings
- ❑ **Routing need not be symmetrical**