NAME

lft − display the route packets take to a network host/socket; optionally show heuristic network information in transitu

SYNOPSIS

lft [**−d** *dport*] [**−s** *sport*] [**−m** *retry min*] [**−M** *retry max*] [**−a** *ahead*]
[**−c** *scatter ms*] [**−t** *timeout ms*] [**−l** *min ttl*] [**−H** *max ttl*] [**−q** *ISN*]
[**−D** *device*] [**−ACENRSTVehinrvz**] *[<gateway> <...>] target:dport*

DESCRIPTION

The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. (from traceroute(8))

lft sends various TCP probes (differing from Van Jacobson's UDP-based method) utilizing the IP protocol 'time to live' field and attempts to elicit an ICMP `TIME_EXCEEDED` (during transit) response from each gateway along the path to some host. **lft** also listens for various TCP and ICMP messages along the way to assist network managers in ascertaining per-protocol heuristic routing information and can optionally retrieve various information about the networks it traverses.

The only mandatory parameter is the target host name or IP number. Options toggle the display of more interesting data or change the variables of the trace itself. The (-E/-e) adaptive option tries several combinations of TCP states (changing flags inside the probes it sends) in order to improve the chances of a successful trace and expose stateful packet filters.

Other options are:

−d *dport*
Set *dport* as the destination TCP port of the probes LFT generates. Default is 80. This option is useful to see if packets follow a different route based on protocol destination, a likely scenario when load balancers or proxies are involved. This option may also bypass less sophisticated packet filter configurations.

−s *sport*
Set *sport* as the origin TCP port of the probes LFT generates. Default is 53. This option is useful to see if packets follow a different route based on protocol source. This option may also bypass less sophisticated packet filter configurations.

−z Automatically select a pseudo-random source port. This option may be useful if your local packet filter or proxy doesn't allow you to use source ports outside of the dymanic range allocation.

−m *min*
Set *min* as the minimum number of probes to send per host. Default is 1 unless adaptive (-E) mode is used.

−M *max*
Set *max* as the maximum number of probes to send per host. Default is 5.

−a *ahead*
Set *ahead* as the number of hops forward to query before waiting for a response. Default is 5.

−c *scatter ms*
Set *scatter ms* as the minimum number of milliseconds to wait between sending probes. Default is 20.

-t *timeout ms*

Set *timeout ms* as the maximum number of milliseconds to wait before assuming a probe was lost/discarded. Default is 1000.

-l *min ttl*

Set *min tll* as the minimum TTL (time-to-live) on outgoing probes (essentially, the first hop in the line that you want to display). Default is 1.

-q *ISN*

Set *ISN* as the ISN (initial sequence number) of the first probe. If unset, one will be automatically generated using a pseudo-random, time-seeded algorithm.

-D *device*

Set *device* as the network device or IP address to be used. (e.g., "en1" or "1.2.3.4") If unset, **lft** will attempt to determine and acquire the appropriate interface based on routing.

-H *ttl*

Set *ttl* as the maximum TTL, essentially the maximum route traversal distance in hops. Default is 30.

-i Disable "stop" on ICMP other than TTL expired.

-n Print addresses numerically rather than symbolically and numerically. Disables use of the DNS resolver completely.

-h Print addresses symbolically rather than symbolically and numerically. If the DNS resolver fails to resolve an address, the address is printed numerically.

-E/e Enable use of the adaptive engine which tries several combinations of TCP states (changing flags inside the probes it sends) in order to improve the chances of a successful trace. The engine also displays other useful information such as stateful inspection firewalls or broken IP stacks encountered along the way.

-N Enable lookup and display of network names (e.g., [GNTY-NETBLK-4]). This option queries various registries of network address allocation such as ARIN, RIPE, and APNIC.

-A Enable lookup and display of of AS (autonomous system) numbers (e.g., [1]). This option queries one of several whois servers (see options 'C' and 'R') in order to ascertain the origin ASN of the IP address in question. By default, LFT uses the pWhoIs service whose ASN data tends to be more accurate and more timely than using the RADB as it is derived from the Internet's global routing table and multiple Tier-1 ISP perspectives. See www.pwhois.org

-r Force use of the RIPE NCC RIS whois service to lookup ASNs. This is an alternative source of timely ASN-related information built using the Internet's global routing table and multiple Tier-1 ISP perspectives. See www.ripe.net/projects/ris

-C Force use of the Cymru whois service to lookup ASNs. This is an alternative source of timely ASN-related information built using the Internet's global routing table and multiple Tier-1 ISP perspectives. See www.cymru.com

-R Force use of the RADB whois service to lookup ASNs. This tends to be quick, but incomplete and usually inaccurate with regard to the 'actual' Internet routing table. See www.radb.net

-T Enable display of LFT's execution timer. This option places timers on the trace itself and on lookups and name resolution to show where LFT is spending its time, waiting on resolvers, or processing trace packets.

−S Suppress display of the real-time status bar. This option makes LFT show its completed trace output only, no-frills.

−V Display verbose output. Use more V's for more info.

−v Display version information, then exit(1).

Any hosts listed after these options and before the final host/target will comprise the loose source route. Since network operators have security concerns regarding the use of source routing, don't expect the LSRR options to do anything for you in most public networks.

EXAMPLES

A sample use and output might be:

```
[edge.lax]$ lft -S 4.2.2.2

Hop  LFT trace to vnsc-bak.sys.gtei.net (4.2.2.2):80/tcp
 1   ln-gateway.centergate.com (206.117.161.1) 0.5ms
 2   isi-acg.ln.net (130.152.136.1) 2.3ms
 3   isi-1-lngw2-atm.ln.net (130.152.180.21) 2.5ms
 4   gigabitethernet5-0.lsanca1-cr3.bbnplanet.net (4.24.4.249) 3.0ms
 5   p6-0.lsanca1-cr6.bbnplanet.net (4.24.4.2) 3.4ms
 6   p6-0.lsanca2-br1.bbnplanet.net (4.24.5.49) 3.3ms
 7   p15-0.snjpca1-br1.bbnplanet.net (4.24.5.58) 10.9ms
 8   so-3-0-0.mtvwca1-br1.bbnplanet.net (4.24.7.33) 11.1ms
 9   p7-0.mtvwca1-dc-dbe1.bbnplanet.net (4.24.9.166) 11.0ms
10   vlan40.mtvwca1-dc1-dfa1-rc1.bbnplanet.net (128.11.193.67) 11.1ms
**   [neglected] no reply packets received from TTLs 11 through 20
**   [4.2-3 BSD bug] the next gateway may errantly reply with reused TTLs
21   [target] vnsc-bak.sys.gtei.net (4.2.2.2) 11.2ms
```

The (-S) option was used to suppress the real-time status bar for clean output. LFT's "**" notifiers in between hops 10 and 21 represent additional useful information: the first is a "[neglected]" indicator that lets us know that none of the probes sent with the TTLs indicated elicited responses. This could be for a variety of reasons, but the cause of this specific occurrence is described in the next informative message which indicates that this is likely the result of a bug in the 4.[23] BSD network code (and its derivatives): BSD 4.x (x < 3) sends an unreachable message using whatever TTL remains in the original datagram. Since, for gateways, the remaining TTL is zero, the ICMP "time exceeded" is guaranteed to not make it back to us. LFT does its best to identify this condition rather than print lots and lots of hops that don't exist (trying to reach a high enough TTL).

Now, using the adaptive engine option:

```
[edge.lax]$ lft -E -S 4.2.2.1

Hop  LFT trace to vnsc-pri.sys.gtei.net (4.2.2.1):80/tcp
 1   ln-gateway.centergate.com (206.117.161.1) 0.5/0.5ms
 2   isi-acg.ln.net (130.152.136.1) 2.1/2.3ms
 3   isi-1-lngw2-atm.ln.net (130.152.180.21) 2.6/7.1ms
 4   gigabitethernet5-0.lsanca1-cr3.bbnplanet.net (4.24.4.249) 6.1/3.9ms
**   [firewall] the next gateway may statefully inspect packets
 5   p0-0-0.lsanca1-csr1.bbnplanet.net (4.24.4.10) 155.4/3.7ms
 6   [target] vnsc-pri.sys.gtei.net (4.2.2.1) 22.6/3.7/*/*/*/*/*ms
```

In the scenario above, the adaptive engine was able to identify a stateful, packet-inspecting firewall in the path. Another example with more options:

```
[edge.lax]$ lft -S -A -T -m 2 -d 80 -s 53 www.yahoo.com

Hop  LFT trace to w9.scd.yahoo.com (66.218.71.88):80/tcp
 1   [226] ln-gateway.centergate.com (206.117.161.1)  1 ms
 2   [226] isi-acg.ln.net (130.152.136.1)  2 ms
 3   [226] isi-1-lngw2-atm.ln.net (130.152.180.21)  3 ms
 4   [1] gigether5-0.lsanca1-cr3.bbnplanet.net (4.24.4.249)  3 ms
 5   [1] p6-0.lsanca1-cr6.bbnplanet.net (4.24.4.2)  5 ms
 6   [1] p6-0.lsanca2-br1.bbnplanet.net (4.24.5.49)  3 ms
 7   [1] p1-0.lsanca2-cr2.bbnplanet.net (4.25.112.1)  3 ms
 8   [16852] pos4-0.core1.LosAngeles1.Level3.net (209.0.227.57)  3 ms
 9   [3356] so-4-0-0.mp1.LosAngeles1.Level3.net (209.247.10.193)  3 ms
10   [3356] so-3-0-0.mp2.SanJose1.Level3.net (64.159.1.130)  11 ms
11   [3356] gige10-0.ipcolo4.SanJose1.Level3.net (64.159.2.42)  11 ms
12   [3356] cust-int.level3.net (64.152.81.62)  52 ms
13   [10310] vl17.bas2.scd.yahoo.com (66.218.64.150)  53 ms
14   [10310] w9.scd.yahoo.com (66.218.71.88) [target]  54 ms

LFT’s trace took 5.23 seconds.  Resolution required 3.58 seconds.
```

Note the -Ar above displays ASNs using the RADB as a whois source. A better option may have been to use the -A alone or perhaps -AC.

And why not request netblock lookups?

```
[edge.lax]$ lft -S -N www.microsoft.com

Hop  LFT trace to www.us.microsoft.com (207.46.197.113):80/tcp
 1   [LOS-NETTOS-BLK4] ln-gateway.centergate.com (206.117.161.1)  2 ms
 2   [LOS-NETTOS] isi-acg.ln.net (130.152.136.1)  3 ms
 3   [LOS-NETTOS] isi-1-lngw2-pos.ln.net (130.152.80.30)  5 ms
 4   [GNTY-4-0] gigether5-0.lsanca1-cr3.bbnplanet.net (4.24.4.249)  4 ms
 5   [GNTY-4-0] p6-0.lsanca1-cr6.bbnplanet.net (4.24.4.2)  3 ms
 6   [GNTY-4-0] p6-0.lsanca2-br1.bbnplanet.net (4.24.5.49)  3 ms
 7   [GNTY-4-0] p15-0.snjpca1-br1.bbnplanet.net (4.24.5.58)  10 ms
 8   [GNTY-4-0] p9-0.snjpca1-br2.bbnplanet.net (4.24.9.130)  11 ms
 9   [GNTY-4-0] so-1-0-0.sttlwa2-br1.bbnplanet.net (4.0.3.229)  27 ms
10   [GNTY-4-0] so-0-0-0.sttlwa1-hcr1.bbnplanet.net (4.24.11.202)  28 ms
11   [GNTY-4-0] so-7-0-0.sttlwa1-hcr2.bbnplanet.net (4.24.10.234)  28 ms
12   [GNTY-4-0] p1-0.sttlwa1-cr2.bbnplanet.net (4.24.10.241)  29 ms
13   [GNTY-4-0] p2-0.msseattle.bbnplanet.net (4.25.89.6)  32 ms
14   [MICROSOFT-GLOBAL-NET] 207.46.154.9  32 ms
15   [MICROSOFT-GLOBAL-NET] 207.46.155.17  33 ms
16   [MICROSOFT-GLOBAL-NET] 207.46.129.51 [prohibited]  35 ms
```

AUTHORS

Victor Oppleman, Eugene Antsilevitch, and other helpers around the world.

FORMER AUTHORS

Nils McCarthy: Thanks to Nils for writing ’FFT’, LFT’s predecessor.

REPORTING BUGS

To report bugs, send e-mail to <lft@oppleman.com>

SEE ALSO

`traceroute`(8), `netstat`(1), `whois`(1), `whob`(8)

HISTORY

The **`lft`** command first appeared in 1998 as ’fft’. Renamed as a result of confusion with fast fourier transforms, **`lft`** stands for ’layer four traceroute.’