[**每天一个linux命令（55）：traceroute命令**](http://www.cnblogs.com/peida/archive/2013/03/07/2947326.html)

通过traceroute我们可以知道信息从你的计算机到互联网另一端的主机是走的什么路径。当然每次数据包由某一同样的出发点（source）到达某一同样的目的地(destination)走的路径可能会不一样，但基本上来说大部分时候所走的路由是相同的。linux系统中，我们称之为traceroute,在MS Windows中为tracert。 traceroute通过发送小的数据包到目的设备直到其返回，来测量其需要多长时间。一条路径上的每个设备traceroute要测3次。输出结果中包括每次测试的时间(ms)和设备的名称（如有的话）及其IP地址。

在大多数情况下，我们会在linux主机系统下，直接执行命令行：

traceroute hostname

而在Windows系统下是执行tracert的命令：

tracert hostname

**1.命令格式：**

traceroute[参数][主机]

**2.命令功能：**

traceroute指令让你追踪网络数据包的路由途径，预设数据包大小是40Bytes，用户可另行设置。

具体参数格式：traceroute [-dFlnrvx][-f<存活数值>][-g<网关>...][-i<网络界面>][-m<存活数值>][-p<通信端口>][-s<来源地址>][-t<服务类型>][-w<超时秒数>][主机名称或IP地址][数据包大小]

**3.命令参数：**

-d 使用Socket层级的排错功能。

-f 设置第一个检测数据包的存活数值TTL的大小。

-F 设置勿离断位。

-g 设置来源路由网关，最多可设置8个。

-i 使用指定的网络界面送出数据包。

-I 使用ICMP回应取代UDP资料信息。

-m 设置检测数据包的最大存活数值TTL的大小。

-n 直接使用IP地址而非主机名称。

-p 设置UDP传输协议的通信端口。

-r 忽略普通的Routing Table，直接将数据包送到远端主机上。

-s 设置本地主机送出数据包的IP地址。

-t 设置检测数据包的TOS数值。

-v 详细显示指令的执行过程。

-w 设置等待远端主机回报的时间。

-x 开启或关闭数据包的正确性检验。

**4.使用实例：**

**实例1：**traceroute 用法简单、最常用的用法

**命令：**

traceroute www.baidu.com

**输出：**

[root@localhost ~]# traceroute www.baidu.com  
traceroute to www.baidu.com (61.135.169.125), 30 hops max, 40 byte packets  
 1  192.168.74.2 (192.168.74.2)  2.606 ms  2.771 ms  2.950 ms  
 2  211.151.56.57 (211.151.56.57)  0.596 ms  0.598 ms  0.591 ms  
 3  211.151.227.206 (211.151.227.206)  0.546 ms  0.544 ms  0.538 ms  
 4  210.77.139.145 (210.77.139.145)  0.710 ms  0.748 ms  0.801 ms  
 5  202.106.42.101 (202.106.42.101)  6.759 ms  6.945 ms  7.107 ms  
 6  61.148.154.97 (61.148.154.97)  718.908 ms \* bt-228-025.bta.net.cn (202.106.228.25)  5.177 ms  
 7  124.65.58.213 (124.65.58.213)  4.343 ms  4.336 ms  4.367 ms  
 8  202.106.35.190 (202.106.35.190)  1.795 ms 61.148.156.138 (61.148.156.138)  1.899 ms  1.951 ms  
 9  \* \* \*  
30  \* \* \*  
[root@localhost ~]#

**说明：**

记录按序列号从1开始，每个纪录就是一跳 ，每跳表示一个网关，我们看到每行有三个时间，单位是 ms，其实就是-q的默认参数。探测数据包向每个网关发送三个数据包后，网关响应后返回的时间；如果您用 traceroute -q 4 www.58.com ，表示向每个网关发送4个数据包。

有时我们traceroute 一台主机时，会看到有一些行是以星号表示的。出现这样的情况，可能是防火墙封掉了ICMP的返回信息，所以我们得不到什么相关的数据包返回数据。

有时我们在某一网关处延时比较长，有可能是某台网关比较阻塞，也可能是物理设备本身的原因。当然如果某台DNS出现问题时，不能解析主机名、域名时，也会 有延时长的现象；您可以加-n 参数来避免DNS解析，以IP格式输出数据。

如果在局域网中的不同网段之间，我们可以通过traceroute 来排查问题所在，是主机的问题还是网关的问题。如果我们通过远程来访问某台服务器遇到问题时，我们用到traceroute 追踪数据包所经过的网关，提交IDC服务商，也有助于解决问题；但目前看来在国内解决这样的问题是比较困难的，就是我们发现问题所在，IDC服务商也不可能帮助我们解决。

**实例2：跳数设置**

**命令：**

traceroute -m 10 www.baidu.com

**输出：**

[root@localhost ~]# traceroute -m 10 www.baidu.com  
traceroute to www.baidu.com (61.135.169.105), 10 hops max, 40 byte packets  
 1  192.168.74.2 (192.168.74.2)  1.534 ms  1.775 ms  1.961 ms  
 2  211.151.56.1 (211.151.56.1)  0.508 ms  0.514 ms  0.507 ms  
 3  211.151.227.206 (211.151.227.206)  0.571 ms  0.558 ms  0.550 ms  
 4  210.77.139.145 (210.77.139.145)  0.708 ms  0.729 ms  0.785 ms  
 5  202.106.42.101 (202.106.42.101)  7.978 ms  8.155 ms  8.311 ms  
 6  bt-228-037.bta.net.cn (202.106.228.37)  772.460 ms bt-228-025.bta.net.cn (202.106.228.25)  2.152 ms 61.148.154.97 (61.148.154.97)  772.107 ms  
 7  124.65.58.221 (124.65.58.221)  4.875 ms 61.148.146.29 (61.148.146.29)  2.124 ms 124.65.58.221 (124.65.58.221)  4.854 ms  
 8  123.126.6.198 (123.126.6.198)  2.944 ms 61.148.156.6 (61.148.156.6)  3.505 ms 123.126.6.198 (123.126.6.198)  2.885 ms  
 9  \* \* \*  
10  \* \* \*  
[root@localhost ~]#

**说明：**

**实例3：显示IP地址，不查主机名**

**命令：**

traceroute -n www.baidu.com

**输出：**

[root@localhost ~]# traceroute -n www.baidu.com  
traceroute to www.baidu.com (61.135.169.125), 30 hops max, 40 byte packets  
 1  211.151.74.2  5.430 ms  5.636 ms  5.802 ms  
 2  211.151.56.57  0.627 ms  0.625 ms  0.617 ms  
 3  211.151.227.206  0.575 ms  0.584 ms  0.576 ms  
 4  210.77.139.145  0.703 ms  0.754 ms  0.806 ms  
 5  202.106.42.101  23.683 ms  23.869 ms  23.998 ms  
 6  202.106.228.37  247.101 ms \* \*  
 7  61.148.146.29  5.256 ms 124.65.58.213  4.386 ms  4.373 ms  
 8  202.106.35.190  1.610 ms 61.148.156.138  1.786 ms 61.148.3.34  2.089 ms  
 9  \* \* \*  
30  \* \* \*  
[root@localhost ~]# traceroute www.baidu.com  
traceroute to www.baidu.com (61.135.169.125), 30 hops max, 40 byte packets  
 1  211.151.74.2 (211.151.74.2)  4.671 ms  4.865 ms  5.055 ms  
 2  211.151.56.57 (211.151.56.57)  0.619 ms  0.618 ms  0.612 ms  
 3  211.151.227.206 (211.151.227.206)  0.620 ms  0.642 ms  0.636 ms  
 4  210.77.139.145 (210.77.139.145)  0.720 ms  0.772 ms  0.816 ms  
 5  202.106.42.101 (202.106.42.101)  7.667 ms  7.910 ms  8.012 ms  
 6  bt-228-025.bta.net.cn (202.106.228.25)  2.965 ms  2.440 ms 61.148.154.97 (61.148.154.97)  431.337 ms  
 7  124.65.58.213 (124.65.58.213)  5.134 ms  5.124 ms  5.044 ms  
 8  202.106.35.190 (202.106.35.190)  1.917 ms  2.052 ms  2.059 ms  
 9  \* \* \*  
30  \* \* \*  
[root@localhost ~]#

**说明：**

**实例4：探测包使用的基本UDP端口设置6888**

**命令：**

traceroute -p 6888 www.baidu.com

**输出：**

[root@localhost ~]# traceroute -p 6888 www.baidu.com  
traceroute to www.baidu.com (220.181.111.147), 30 hops max, 40 byte packets  
 1  211.151.74.2 (211.151.74.2)  4.927 ms  5.121 ms  5.298 ms  
 2  211.151.56.1 (211.151.56.1)  0.500 ms  0.499 ms  0.509 ms  
 3  211.151.224.90 (211.151.224.90)  0.637 ms  0.631 ms  0.641 ms  
 4  \* \* \*  
 5  220.181.70.98 (220.181.70.98)  5.050 ms  5.313 ms  5.596 ms  
 6  220.181.17.94 (220.181.17.94)  1.665 ms !X \* \*  
[root@localhost ~]#

**说明：**

**实例5：把探测包的个数设置为值4**

**命令：**

traceroute -q 4 www.baidu.com

**输出：**

[root@localhost ~]# traceroute -q 4 www.baidu.com  
traceroute to www.baidu.com (61.135.169.125), 30 hops max, 40 byte packets  
 1  211.151.74.2 (211.151.74.2)  40.633 ms  40.819 ms  41.004 ms  41.188 ms  
 2  211.151.56.57 (211.151.56.57)  0.637 ms  0.633 ms  0.627 ms  0.619 ms  
 3  211.151.227.206 (211.151.227.206)  0.505 ms  0.580 ms  0.571 ms  0.569 ms  
 4  210.77.139.145 (210.77.139.145)  0.753 ms  0.800 ms  0.853 ms  0.904 ms  
 5  202.106.42.101 (202.106.42.101)  7.449 ms  7.543 ms  7.738 ms  7.893 ms  
 6  61.148.154.97 (61.148.154.97)  316.817 ms bt-228-025.bta.net.cn (202.106.228.25)  3.695 ms  3.672 ms \*  
 7  124.65.58.213 (124.65.58.213)  3.056 ms  2.993 ms  2.960 ms 61.148.146.29 (61.148.146.29)  2.837 ms  
 8  61.148.3.34 (61.148.3.34)  2.179 ms  2.295 ms  2.442 ms 202.106.35.190 (202.106.35.190)  7.136 ms  
 9  \* \* \* \*  
30  \* \* \* \*  
[root@localhost ~]#

**说明：**

**实例6：绕过正常的路由表，直接发送到网络相连的主机**

**命令：**

 traceroute -r www.baidu.com

**输出：**

[root@localhost ~]# traceroute -r www.baidu.com  
traceroute to www.baidu.com (61.135.169.125), 30 hops max, 40 byte packets  
connect: 网络不可达  
[root@localhost ~]#

**说明：**

**实例7：**把对外发探测包的等待响应时间设置为3秒

**命令：**

traceroute -w 3 www.baidu.com

**输出：**

[root@localhost ~]# traceroute -w 3 www.baidu.com  
traceroute to www.baidu.com (61.135.169.105), 30 hops max, 40 byte packets  
 1  211.151.74.2 (211.151.74.2)  2.306 ms  2.469 ms  2.650 ms  
 2  211.151.56.1 (211.151.56.1)  0.621 ms  0.613 ms  0.603 ms  
 3  211.151.227.206 (211.151.227.206)  0.557 ms  0.560 ms  0.552 ms  
 4  210.77.139.145 (210.77.139.145)  0.708 ms  0.761 ms  0.817 ms  
 5  202.106.42.101 (202.106.42.101)  7.520 ms  7.774 ms  7.902 ms  
 6  bt-228-025.bta.net.cn (202.106.228.25)  2.890 ms  2.369 ms 61.148.154.97 (61.148.154.97)  471.961 ms  
 7  124.65.58.221 (124.65.58.221)  4.490 ms  4.483 ms  4.472 ms  
 8  123.126.6.198 (123.126.6.198)  2.948 ms 61.148.156.6 (61.148.156.6)  7.688 ms  7.756 ms  
 9  \* \* \*  
30  \* \* \*  
[root@localhost ~]#

**说明：**

**Traceroute的工作原理：**

Traceroute最简单的基本用法是：traceroute hostname

Traceroute程序的设计是利用ICMP及IP header的TTL（Time To Live）栏位（field）。首先，traceroute送出一个TTL是1的IP datagram（其实，每次送出的为3个40字节的包，包括源地址，目的地址和包发出的时间标签）到目的地，当路径上的第一个路由器（router）收到这个datagram时，它将TTL减1。此时，TTL变为0了，所以该路由器会将此datagram丢掉，并送回一个「ICMP time exceeded」消息（包括发IP包的源地址，IP包的所有内容及路由器的IP地址），traceroute 收到这个消息后，便知道这个路由器存在于这个路径上，接着traceroute 再送出另一个TTL是2 的datagram，发现第2 个路由器...... traceroute 每次将送出的datagram的TTL 加1来发现另一个路由器，这个重复的动作一直持续到某个datagram 抵达目的地。当datagram到达目的地后，该主机并不会送回ICMP time exceeded消息，因为它已是目的地了，那么traceroute如何得知目的地到达了呢？

Traceroute在送出UDP datagrams到目的地时，它所选择送达的port number 是一个一般应用程序都不会用的号码（30000 以上），所以当此UDP datagram 到达目的地后该主机会送回一个「ICMP port unreachable」的消息，而当traceroute 收到这个消息时，便知道目的地已经到达了。所以traceroute 在Server端也是没有所谓的Daemon 程式。

Traceroute提取发 ICMP TTL到期消息设备的IP地址并作域名解析。每次 ，Traceroute都打印出一系列数据,包括所经过的路由设备的域名及 IP地址,三个包每次来回所花时间。

**windows之tracert:**

**格式：**

tracert [-d] [-h maximum\_hops] [-j host-list] [-w timeout] target\_name

**参数说明：**

tracert [-d] [-h maximum\_hops] [-j computer-list] [-w timeout] target\_name

该诊断实用程序通过向目的地发送具有不同生存时间 (TL) 的 Internet 控制信息协议 (CMP) 回应报文，以确定至目的地的路由。路径上的每个路由器都要在转发该 ICMP 回应报文之前将其 TTL 值至少减 1，因此 TTL 是有效的跳转计数。当报文的 TTL 值减少到 0 时，路由器向源系统发回 ICMP 超时信息。通过发送 TTL 为 1 的第一个回应报文并且在随后的发送中每次将 TTL 值加 1，直到目标响应或达到最大 TTL 值，Tracert 可以确定路由。通过检查中间路由器发发回的 ICMP 超时 (ime Exceeded) 信息，可以确定路由器。注意，有些路由器“安静”地丢弃生存时间 (TLS) 过期的报文并且对 tracert 无效。

参数：

-d 指定不对计算机名解析地址。

-h maximum\_hops 指定查找目标的跳转的最大数目。

-jcomputer-list 指定在 computer-list 中松散源路由。

-w timeout 等待由 timeout 对每个应答指定的毫秒数。

target\_name 目标计算机的名称。

实例：

C:\Users\Administrator>tracert www.58.com  
  
Tracing route to www.58.com [221.187.111.30]  
over a maximum of 30 hops:  
  
  1     1 ms     1 ms     1 ms  10.58.156.1  
  2     1 ms    <1 ms    <1 ms  10.10.10.1  
  3     1 ms     1 ms     1 ms  211.103.193.129  
  4     2 ms     2 ms     2 ms  10.255.109.129  
  5     1 ms     1 ms     3 ms  124.205.98.205  
  6     2 ms     2 ms     2 ms  124.205.98.253  
  7     2 ms     6 ms     1 ms  202.99.1.125  
  8     5 ms     6 ms     5 ms  118.186.0.113  
  9   207 ms     \*        \*     118.186.0.106  
 10     8 ms     6 ms    11 ms  124.238.226.201  
 11     6 ms     7 ms     6 ms  219.148.19.177  
 12    12 ms    12 ms    16 ms  219.148.18.117  
 13    14 ms    17 ms    16 ms  219.148.19.125  
 14    13 ms    13 ms    12 ms  202.97.80.113  
 15     \*        \*        \*     Request timed out.  
 16    12 ms    12 ms    17 ms  bj141-147-82.bjtelecom.net [219.141.147.82]  
 17    13 ms    13 ms    12 ms  202.97.48.2  
 18     \*        \*        \*     Request timed out.  
 19    14 ms    14 ms    12 ms  221.187.224.85  
 20    15 ms    13 ms    12 ms  221.187.104.2  
 21     \*        \*        \*     Request timed out.  
 22    15 ms    17 ms    18 ms  221.187.111.30  
  
Trace complete.