

网络问题如何排查

典型回答

网络问题是指在典型的C/S或B/S通信中，客户端或浏览器向服务端发送数或接受数据的时候，因为网络不稳定或接收方异常导致的数据包丢失、无法连接等问题。通常情况下TCP会通过重试的机制确保数据包在一段时间后到达接收方。如果是UDP需要应用自身实现重传机制确保在出现丢包后可重新发送。

注意本文讨论的内容基于Linux操作系统

我们可以使用一些基本的命令检测是否出现丢包、目标端口异常：

- `ping` 检测本机到目标主机是否通畅
- `telnet` 检测本机到目标主机端口是否通畅
- `traceroute` 检测本机到目标主机经过的路由是否通畅

扩展知识

分别解释下ping、telnet、traceroute

ping 命令会发出 ICMP 类型的数据包，使用方法：

```
ping 目标ip/域名 # 如果不使用-c参数会一直ping，按ctrl+c停止
```

```
[root@VM-24-16-centos ~]# ping -c 5 baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=251 time=4.80 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=251 time=4.80 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=251 time=4.81 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=4 ttl=251 time=4.82 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=5 ttl=251 time=4.81 ms

--- baidu.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 4.801/4.812/4.826/0.062 ms
[root@VM-24-16-centos ~]# ping -c 5 google.com
PING google.com (172.217.160.78) 56(84) bytes of data.
^C
--- google.com ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 3999ms

[root@VM-24-16-centos ~]#
```

每个包的字节数、序号、TTL、请求来回的耗时

发送5个数据包，接收5个回执包，0%丢包

发送5数据包，接受0个会质保，100%丢包

ping存在的问题：

1. 由于ping使用ICMP协议，一些主机防火墙会丢弃ICMP报文，所以会导致ping不同
2. ping只是探测本机到目标主机的通路，并不探测目标端口是否有效

telnet 目标ip/域名 端口 #telnet本意是连接到目标主机和端口成功后，以交互式发送数据包。所以在探测端口是否正常打开时，我们需要按下ctrl+c，然后输入q退出telnet。否则telnet会一直等待用户的输入。

```
[root@VM-24-16-centos ~]# ping -c 5 baidu.com
PING baidu.com (39.156.66.10) 56(84) bytes of data.
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=1 ttl=251 time=4.80 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=2 ttl=251 time=4.80 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=3 ttl=251 time=4.81 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=4 ttl=251 time=4.82 ms
64 bytes from 39.156.66.10 (39.156.66.10): icmp_seq=5 ttl=251 time=4.81 ms

--- baidu.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 4.801/4.812/4.826/0.062 ms
[root@VM-24-16-centos ~]# ping -c 5 google.com
PING google.com (172.217.160.78) 56(84) bytes of data.
^C
--- google.com ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 3999ms
```

每个包的字节数、序号、TTL、请求来回的耗时

发送5个数据包，接收5个回执包，0%丢包

发送5数据包，接受0个会质保，100%丢包

```
[root@VM-24-16-centos ~]#
```

telnet存在的问题：

1. 无法感知到达目标主机的响应耗时
2. 交互式的体验，不太适合写在shell脚本中

traceroute 目标ip/域名

#traceroute常用参数：

1. -w 秒数 等待远端主机响应的的时间，高于这个时间算为超时
2. -i 网卡 使用指定网卡发出数据包
3. -g 网关ip 设置经过哪个网关发出数据包
4. -m 跳数 设置经过路由的最大跳数

```

[root@VM-24-16-centos ~]# traceroute baidu.com
traceroute to baidu.com (39.156.66.10), 30 hops max, 60 byte packets
 1  11.42.21.129 (11.42.21.129)  0.557 ms  0.972 ms  0.790 ms
 2  11.42.84.34 (11.42.84.34)  0.692 ms  1.243 ms  0.965 ms
 3  10.196.58.5 (10.196.58.5)  13.369 ms  19.391 ms  19.278 ms
 4  10.200.54.201 (10.200.54.201)  1.126 ms  10.200.54.193 (10.200.54.193)  1.097 ms  10.200.54.201 (10.200.54.201)  1.096 ms
 5  39.156.7.182 (39.156.7.182)  2.190 ms  2.385 ms  2.279 ms
 6  39.156.0.81 (39.156.0.81)  2.777 ms  39.156.0.85 (39.156.0.85)  3.927 ms  3.685 ms
 7  111.13.188.38 (111.13.188.38)  4.132 ms  8.536 ms  111.13.0.174 (111.13.0.174)  11.051 ms
 8  39.156.27.1 (39.156.27.1)  4.378 ms  39.156.27.5 (39.156.27.5)  4.537 ms  39.156.27.1 (39.156.27.1)  4.892 ms
 9  39.156.67.17 (39.156.67.17)  4.008 ms  39.156.67.1 (39.156.67.1)  3.191 ms  39.156.67.17 (39.156.67.17)  4.123 ms
10  * * *
11  * * *
12  * * *

```

默认会发3个包，这里显示了3个包各自的响应耗时

包括3个包到达的ip和耗时

无响应会返回***

traceroute主要用于跟踪路由，排查数据包到达目标主机之前产生的问题。

除了基本的三个工具还有什么？

curl：一个命令行下的http客户端，可以通过curl打印整个http请求过程的耗时，方法如下：

1. 以下模板保存到一个文件中，如：curl.txt

```

\n\nquery time:\n-----\n
time_namelookup:  %{time_namelookup}\n
time_connect:    %{time_connect}\n
time_appconnect:  %{time_appconnect}\n
time_redirect:    %{time_redirect}\n
time_pretransfer: %{time_pretransfer}\n
time_starttransfer:  %{time_starttransfer}\n
-----\n
time_total:  %{time_total}\n

```

2. 使用curl 命令发起请求

```
curl -w '@curl.txt' baidu.com -vvv
```

```

root@VM-24-16-centos ~]# curl -w '@curl.txt' baidu.com -vvv
* About to connect() to baidu.com port 80 (#0)
* Trying 110.242.68.66...
* Connected to baidu.com (110.242.68.66) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: baidu.com
> Accept: */*
< HTTP/1.1 200 OK
< Date: Mon, 27 Nov 2023 01:40:29 GMT
< Server: Apache
< Last-Modified: Tue, 12 Jan 2010 13:48:00 GMT
< ETag: "51-47cf7e6ee8400"
< Accept-Ranges: bytes
< Content-Length: 81
< Cache-Control: max-age=86400
< Expires: Tue, 28 Nov 2023 01:40:29 GMT
< Connection: Keep-Alive
< Content-Type: text/html
<
<html>
<meta http-equiv="refresh" content="0;url=http://www.baidu.com/">
</html>
* Connection #0 to host baidu.com left intact

query time:
-----
time_namelookup: 0.004
time_connect: 0.015
time_appconnect: 0.000
time_redirect: 0.000
time_pretransfer: 0.015
time_starttransfer: 0.036
-----
time_total: 0.036

```

http请求头

http响应头

http响应体

各阶段耗时

各阶段响应耗时解释：

- time_namelookup: 域名解析所需的时间。
- time_connect: TCP 连接建立所需的时间。

- time_appconnect: TLS/SSL 握手完成所需的时间。
- time_pretransfer: 从请求开始到响应开始传输的时间。
- time_redirect: 重定向所需的时间。
- time_starttransfer: 从请求开始到第一个字节将要传输的时间，包括预传输时间和服务器处理结果所需的时间。
- time_total: 整个请求过程所需的总时间。

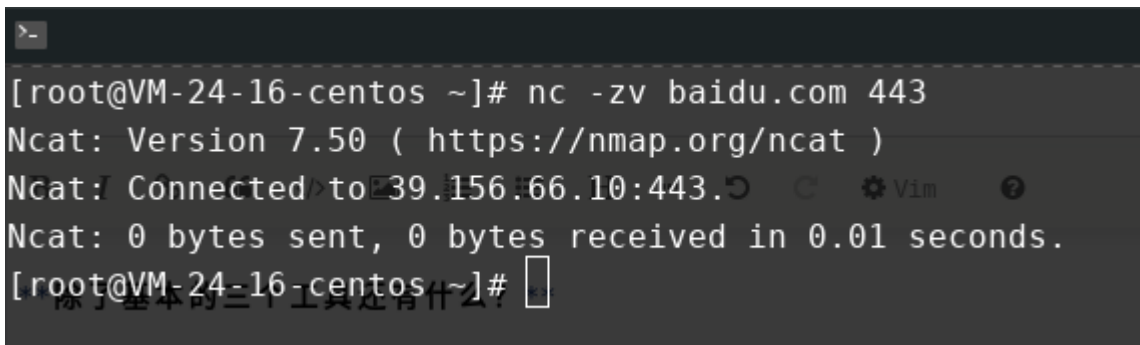
tcpping: tcpping是一个第三方的ping工具，可以用来ping某个端口。ping使用ICMP协议，而tcpping是通过TCP协议向给定的端口建立连接。**注意该工具需要单独安装。**

```
yum install tcptraceroute -y #安装依赖
wget http://www.vdberg.org/~richard/tcpping #下载tcpping
chmod 755 tcpping #设置可执行权限
./tcpping www.baidu.com 443
```

```
[root@VM-24-16-centos ~]# ./tcpping baidu.com 443
traceroute to baidu.com (39.156.66.10), 255 hops max, 60 byte packets
seq 0: tcp response from 39.156.66.10 (39.156.66.10) <syn,ack> 4.622 ms
traceroute to baidu.com (39.156.66.10), 255 hops max, 60 byte packets
seq 1: tcp response from 39.156.66.10 (39.156.66.10) <syn,ack> 4.268 ms
traceroute to baidu.com (110.242.68.66), 255 hops max, 60 byte packets
seq 2: tcp response from 110.242.68.66 (110.242.68.66) <syn,ack> 10.515 ms
traceroute to baidu.com (39.156.66.10), 255 hops max, 60 byte packets
seq 3: tcp response from 39.156.66.10 (39.156.66.10) <syn,ack> 4.300 ms
traceroute to baidu.com (39.156.66.10), 255 hops max, 60 byte packets
seq 4: tcp response from 39.156.66.10 (39.156.66.10) <syn,ack> 4.804 ms
^Cme_connect:  %{time_connect}\n
[root@VM-24-16-centos ~]#me_connect)\n
time_redirect:  %{time_redirect}\n
time_pretransfer:  %{time_pretransfer}\n
time_starttransfer:  %{time_starttransfer}\n
```

当然如果你不想这么复杂也可以使用 nc 命令，通常的Linux发行版都会自带这个命令

```
nc -zv www.baidu.com 443
```

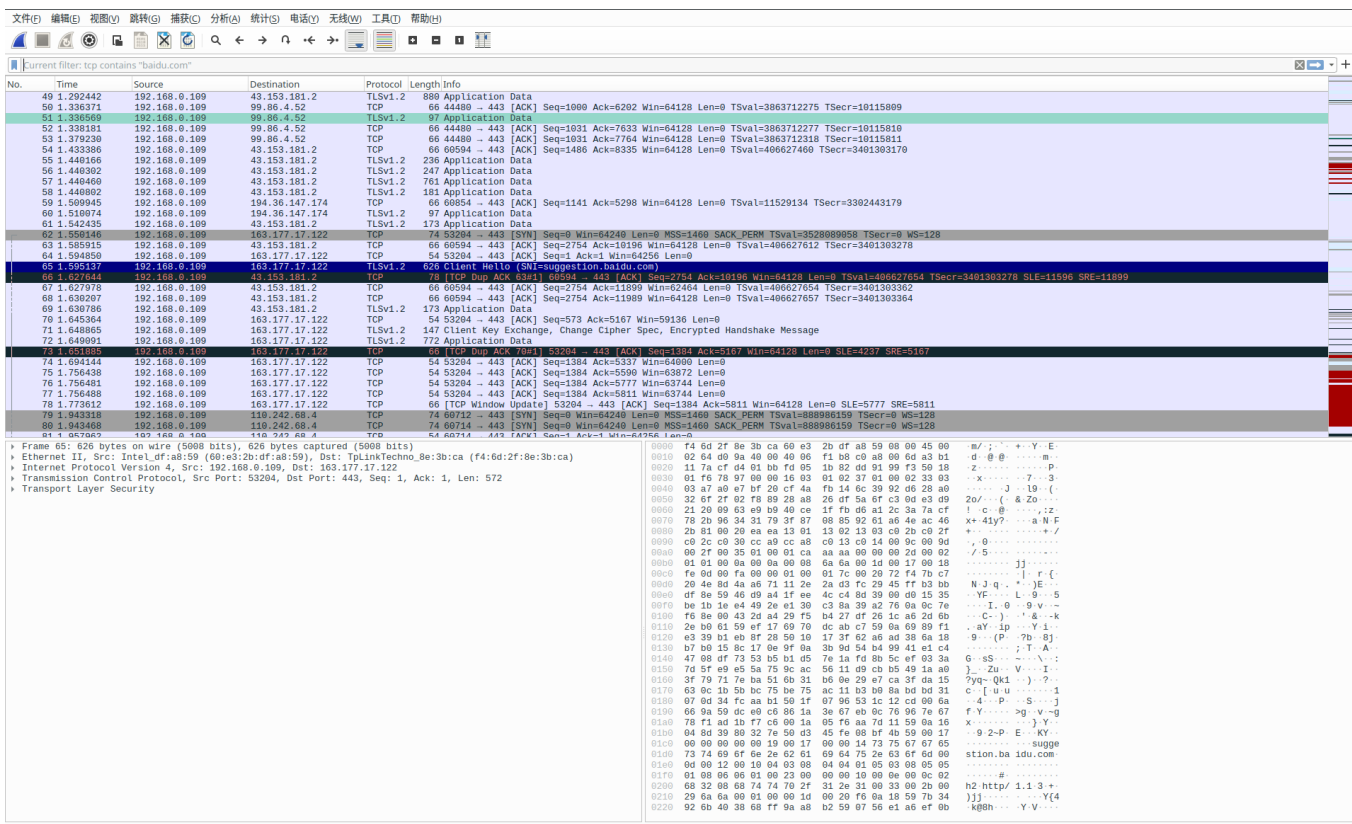


tcpdump: tcpdump是很强大的命令行抓包工具，可以抓各种协议的数据包，同时配合wireshark图形化工具来分析网络通信。

tcpdump在一些Linux发行版中可能未安装，需要手动安装。wireshark可以在 [wireshark官网](#) 进行下载

```
tcpmdump -i eth0 dst port 443 -w 1.cap #抓取网卡eth0 目标端口为443的数据包保存到1.cap文件中
```

通过wireshark打开1.cap进行分析



如何抓包