

34 | 关于 Linux 网络，你必须知道这些（下）

2019-02-08 倪朋飞



讲述：冯永吉

时长 09:55 大小 9.09M



你好，我是倪朋飞。

上一节，我带你学习了 Linux 网络的基础原理。简单回顾一下，Linux 网络根据 TCP/IP 模型，构建其网络协议栈。TCP/IP 模型由应用层、传输层、网络层、网络接口层等四层组成，这也是 Linux 网络栈最核心的构成部分。

应用程序通过套接字接口发送数据包时，先要在网络协议栈中从上到下逐层处理，然后才最终送到网卡发送出去；而接收数据包时，也要先经过网络栈从下到上的逐层处理，最后送到应用程序。

了解 Linux 网络的基本原理和收发流程后，你肯定迫不及待想知道，如何去观察网络的性能情况。具体而言，哪些指标可以用来衡量 Linux 的网络性能呢？

性能指标

实际上，我们通常用带宽、吞吐量、延时、PPS (Packet Per Second) 等指标衡量网络的性能。

带宽，表示链路的最大传输速率，单位通常为 b/s (比特 / 秒)。

吞吐量，表示单位时间内成功传输的数据量，单位通常为 b/s (比特 / 秒) 或者 B/s (字节 / 秒)。吞吐量受带宽限制，而吞吐量 / 带宽，也就是该网络的使用率。

延时，表示从网络请求发出后，一直到收到远端响应，所需要的时间延迟。在不同场景中，这一指标可能会有不同含义。比如，它可以表示，建立连接需要的时间 (比如 TCP 握手延时)，或一个数据包往返所需的时间 (比如 RTT)。

PPS，是 Packet Per Second (包 / 秒) 的缩写，表示以网络包为单位的传输速率。PPS 通常用来评估网络的转发能力，比如硬件交换机，通常可以达到线性转发 (即 PPS 可以达到或者接近理论最大值)。而基于 Linux 服务器的转发，则容易受网络包大小的影响。

除了这些指标，**网络的可用性** (网络能否正常通信)、**并发连接数** (TCP 连接数量)、**丢包率** (丢包百分比)、**重传率** (重新传输的网络包比例) 等也是常用的性能指标。


接下来，请你打开一个终端，SSH 登录到服务器上，然后跟我一起来探索、观测这些性能指标。

网络配置

分析网络问题的第一步，通常是查看网络接口的配置和状态。你可以使用 `ifconfig` 或者 `ip` 命令，来查看网络的配置。我个人更推荐使用 `ip` 工具，因为它提供了更丰富的功能和更易用的接口。

`ifconfig` 和 `ip` 分别属于软件包 `net-tools` 和 `iproute2`，`iproute2` 是 `net-tools` 的下一代。通常情况下它们会在发行版中默认安装。但如果你找不到 `ifconfig` 或者 `ip` 命令，可以安装这两个软件包。

以网络接口 `eth0` 为例，你可以运行下面的两个命令，查看它的配置和状态：

 复制代码

```
1 $ ifconfig eth0
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
3      inet 10.240.0.30 netmask 255.240.0.0 broadcast 10.255.255.255
4      inet6 fe80::20d:3aff:fe07:cf2a prefixlen 64 scopeid 0x20<link>
5      ether 78:0d:3a:07:cf:3a txqueuelen 1000 (Ethernet)
6      RX packets 40809142 bytes 9542369803 (9.5 GB)
7      RX errors 0 dropped 0 overruns 0 frame 0
8      TX packets 32637401 bytes 4815573306 (4.8 GB)
9      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 $ ip -s addr show dev eth0
12 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
13   link/ether 78:0d:3a:07:cf:3a brd ff:ff:ff:ff:ff:ff
14   inet 10.240.0.30/12 brd 10.255.255.255 scope global eth0
15       valid_lft forever preferred_lft forever
16   inet6 fe80::20d:3aff:fe07:cf2a/64 scope link
17       valid_lft forever preferred_lft forever
18   RX: bytes  packets  errors  dropped  overrun  mcast
19   9542432350 40809397 0      0      0      193
20   TX: bytes  packets  errors  dropped  carrier  collsns
21   4815625265 32637658 0      0      0      0
```

你可以看到，`ifconfig` 和 `ip` 命令输出的指标基本相同，只是显示格式略微不同。比如，它们都包括了网络接口的状态标志、MTU 大小、IP、子网、MAC 地址以及网络包收发的统计信息。

这些具体指标的含义，在文档中都有详细的说明，不过，这里有几个跟网络性能密切相关的指标，需要你特别关注一下。

第一，网络接口的状态标志。`ifconfig` 输出中的 `RUNNING`，或 `ip` 输出中的 `LOWER_UP`，都表示物理网络是连通的，即网卡已经连接到了交换机或者路由器中。如果你看不到它们，通常表示网线被拔掉了。

第二，MTU 的大小。MTU 默认大小是 1500，根据网络架构的不同（比如是否使用了 VXLAN 等叠加网络），你可能需要调大或者调小 MTU 的数值。

第三，网络接口的 IP 地址、子网以及 MAC 地址。这些都是保障网络功能正常工作所必需的，你需要确保配置正确。

第四，网络收发的字节数、包数、错误数以及丢包情况，特别是 TX 和 RX 部分的 `errors`、`dropped`、`overruns`、`carrier` 以及 `collisions` 等指标不为 0 时，通常表示出现了网络 I/O 问题。其中：

errors 表示发生错误的数据包数，比如校验错误、帧同步错误等；

dropped 表示丢弃的数据包数，即数据包已经收到了 Ring Buffer，但因为内存不足等原因丢包；

overruns 表示超限数据包数，即网络 I/O 速度过快，导致 Ring Buffer 中的数据包来不及处理（队列满）而导致的丢包；

carrier 表示发生 carrier 错误的数据包数，比如双工模式不匹配、物理电缆出现问题等；


collisions 表示碰撞数据包数。

套接字信息

ifconfig 和 ip 只显示了网络接口收发数据包的统计信息，但在实际的性能问题中，网络协议栈中的统计信息，我们也必须关注。你可以用 netstat 或者 ss，来查看套接字、网络栈、网络接口以及路由表的信息。

我个人更推荐，使用 ss 来查询网络的连接信息，因为它比 netstat 提供了更好的性能（速度更快）。

比如，你可以执行下面的命令，查询套接字信息：

 复制代码

```
1 # head -n 3 表示只显示前面 3 行
2 # -l 表示只显示监听套接字
3 # -n 表示显示数字地址和端口（而不是名字）
4 # -p 表示显示进程信息
5 $ netstat -nlp | head -n 3
6 Active Internet connections (only servers)
7 Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Prog
8 tcp      0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      840/syst
9
10 # -l 表示只显示监听套接字
11 # -t 表示只显示 TCP 套接字
12 # -n 表示显示数字地址和端口（而不是名字）
13 # -p 表示显示进程信息
14 $ ss -ltnp | head -n 3
15 State      Recv-Q    Send-Q      Local Address:Port      Peer Address:Port
16 LISTEN     0          128         127.0.0.53%lo:53        0.0.0.0:*           users:(
17 LISTEN     0          128         0.0.0.0:22              0.0.0.0:*           users:(
```

netstat 和 ss 的输出也是类似的，都展示了套接字的状态、接收队列、发送队列、本地地址、远端地址、进程 PID 和进程名称等。

其中，接收队列（Recv-Q）和发送队列（Send-Q）需要你特别关注，它们通常应该是 0。当你发现它们不是 0 时，说明有网络包的堆积发生。当然还要注意，在不同套接字状态下，它们的含义不同。

当套接字处于连接状态（Established）时，

Recv-Q 表示套接字缓冲还没有被应用程序取走的字节数（即接收队列长度）。

而 Send-Q 表示还没有被远端主机确认的字节数（即发送队列长度）。

当套接字处于监听状态（Listening）时，

Recv-Q 表示 syn backlog 的当前值。

而 Send-Q 表示最大的 syn backlog 值。


而 syn backlog 是 TCP 协议栈中的半连接队列长度，相应的也有一个全连接队列（accept queue），它们都是维护 TCP 状态的重要机制。

顾名思义，所谓半连接，就是还没有完成 TCP 三次握手的连接，连接只进行了一半，而服务器收到了客户端的 SYN 包后，就会把这个连接放到半连接队列中，然后再向客户端发送 SYN+ACK 包。

而全连接，则是指服务器收到了客户端的 ACK，完成了 TCP 三次握手，然后就会把这个连接挪到全连接队列中。这些全连接中的套接字，还需要再被 accept() 系统调用取走，这样，服务器就可以开始真正处理客户端的请求了。

协议栈统计信息

类似的，使用 netstat 或 ss，也可以查看协议栈的信息：

 复制代码

```
1 $ netstat -s
2 ...
3 Tcp:
```

```

4      3244906 active connection openings
5      23143 passive connection openings
6      115732 failed connection attempts
7      2964 connection resets received
8      1 connections established
9      13025010 segments received
10     17606946 segments sent out
11     44438 segments retransmitted
12     42 bad segments received
13     5315 resets sent
14     InCsumErrors: 42
15 ...
16
17 $ ss -s
18 Total: 186 (kernel 1446)
19 TCP:    4 (estab 1, closed 0, orphaned 0, synrecv 0, timewait 0/0), ports 0
20
21 Transport Total      IP        IPv6
22 *          1446      -         -
23 RAW         2         1         1
24 UDP         2         2         0
25 TCP         4         3         1
26 ...

```


这些协议栈的统计信息都很直观。ss 只显示已经连接、关闭、孤儿套接字等简要统计，而 netstat 则提供的是更详细的网络协议栈信息。

比如，上面 netstat 的输出示例，就展示了 TCP 协议的主动连接、被动连接、失败重试、发送和接收的分段数量等各种信息。

网络吞吐和 PPS

接下来，我们再来看看，如何查看系统当前的网络吞吐量和 PPS。在这里，我推荐使用我们的老朋友 sar，在前面的 CPU、内存和 I/O 模块中，我们已经多次用到它。

给 sar 增加 -n 参数就可以查看网络的统计信息，比如网络接口（DEV）、网络接口错误（EDEV）、TCP、UDP、ICMP 等等。执行下面的命令，你就可以得到网络接口统计信息：

 复制代码

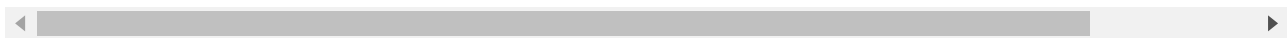
```

1 # 数字 1 表示每隔 1 秒输出一组数据
2 $ sar -n DEV 1

```



```
3 Linux 4.15.0-1035-azure (ubuntu)          01/06/19          _x86_64_          (2 CPU)
4
5 13:21:40          IFACE    rxpck/s    txpck/s    rxkB/s    txkB/s    rxcmp/s    txcmp/s    rxmcs
6 13:21:41          eth0      18.00      20.00      5.79      4.25      0.00      0.00      0
7 13:21:41        docker0      0.00      0.00      0.00      0.00      0.00      0.00      0
8 13:21:41           lo       0.00      0.00      0.00      0.00      0.00      0.00      0
```



这儿输出的指标比较多，我来简单解释下它们的含义。


rxpck/s 和 txpck/s 分别是接收和发送的 PPS，单位为包 / 秒。

rxkB/s 和 txkB/s 分别是接收和发送的吞吐量，单位是 KB/ 秒。

rxcmp/s 和 txcmp/s 分别是接收和发送的压缩数据包数，单位是包 / 秒。

%ifutil 是网络接口的使用率，即半双工模式下为 (rxkB/s+txkB/s)/Bandwidth，而全双工模式下为 max(rxkB/s, txkB/s)/Bandwidth。


其中，Bandwidth 可以用 ethtool 来查询，它的单位通常是 Gb/s 或者 Mb/s，不过注意这里小写字母 b，表示比特而不是字节。我们通常提到的千兆网卡、万兆网卡等，单位也都是比特。如下你可以看到，我的 eth0 网卡就是一个千兆网卡：

 复制代码

```
1 $ ethtool eth0 | grep Speed
2      Speed: 1000Mb/s
```

连通性和延时

最后，我们通常使用 ping，来测试远程主机的连通性和延时，而这基于 ICMP 协议。比如，执行下面的命令，你就可以测试本机到 114.114.114.114 这个 IP 地址的连通性和延时：

 复制代码

```
1 # -c3 表示发送三次 ICMP 包后停止
2 $ ping -c3 114.114.114.114
3 PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.
4 64 bytes from 114.114.114.114: icmp_seq=1 ttl=54 time=244 ms
5 64 bytes from 114.114.114.114: icmp_seq=2 ttl=47 time=244 ms
6 64 bytes from 114.114.114.114: icmp_seq=3 ttl=67 time=244 ms
7
```

```
8 --- 114.114.114.114 ping statistics ---
9 3 packets transmitted, 3 received, 0% packet loss, time 2001ms
10 rtt min/avg/max/mdev = 244.023/244.070/244.105/0.034 ms
```

ping 的输出，可以分为两部分。

第一部分，是每个 ICMP 请求的信息，包括 ICMP 序列号 (icmp_seq)、TTL (生存时间，或者跳数) 以及往返延时。

第二部分，则是三次 ICMP 请求的汇总。

比如上面的示例显示，发送了 3 个网络包，并且接收到 3 个响应，没有丢包发生，这说明测试主机到 114.114.114.114 是连通的；平均往返延时 (RTT) 是 244ms，也就是从发送 ICMP 开始，到接收到 114.114.114.114 回复的确认，总共经历 244ms。

小结

我们通常使用带宽、吞吐量、延时等指标，来衡量网络的性能；相应的，你可以用 ifconfig、netstat、ss、sar、ping 等工具，来查看这些网络的性能指标。

在下一节中，我将以经典的 C10K 和 C100K 问题，带你进一步深入 Linux 网络的工作原理。

思考

最后，我想请你来聊聊，你理解的 Linux 网络性能。你常用什么指标来衡量网络的性能？又用什么思路分析相应性能问题呢？你可以结合今天学到的知识，提出自己的观点。

欢迎在留言区和我讨论，也欢迎你把这篇文章分享给你的同事、朋友。我们一起在实战中演练，在交流中进步。



Linux 性能优化实战

10 分钟帮你找到系统瓶颈

倪朋飞

微软资深工程师
Kubernetes 项目维护者



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得转载

上一篇 33 | 关于 Linux 网络，你必须知道这些（上）

下一篇 35 | 基础篇：C10K 和 C1000K 回顾

精选留言 (15)

写留言



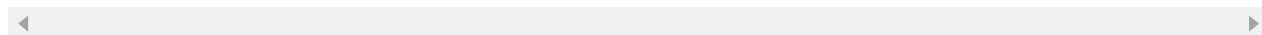
Days

2019-02-09

6

老师春节不休息，大赞啊，老师可否讲解一下一个包从网卡接收，发送在内核协议栈的整个流程，这样性能分析的时候，更好的理解数据包阻塞在哪里？

作者回复：这些在后面的案例中会涉及



于欣磊

2019-03-01

2

小狗同学问到：老师，您好 ss —lntp 这个 当session处于listening中 rec-q 确定是 syn

的backlog吗？

A: Recv-Q为全连接队列当前使用了多少。 中文资料里这个问题讲得最明白的文章：
<https://mp.weixin.qq.com/s/yH3PzGEFopbpA-jw4MythQ>



code2

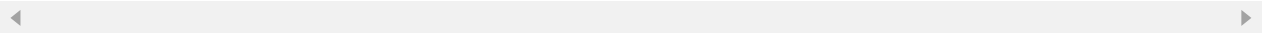
2019-02-10

👍 2

每期读两遍，看看别人怎么做!

展开 ▾

作者回复: 🐣



芥菜

2019-02-08

👍 2

春节期间终于跟上节奏，春节里做到只长知识不长肉：)

展开 ▾



xfan

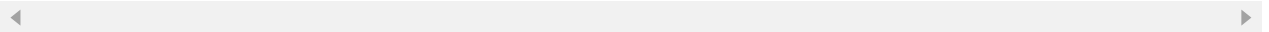
2019-02-11

👍 1

Speed 有的通过ethtool查不到，是什么原因呢，那查不到的话，默认值是多少呢

展开 ▾

作者回复: 可能跟网卡状态和驱动有关，可以试试先配置好 IP 并开启网卡后再查看



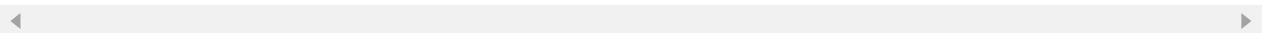
[小狗]

2019-02-09

👍 1

老师，您好 ss —lntp 这个 当session处于listening中 rec-q 确定是 syn的backlog吗？
我之前都是当做全队列的长度

作者回复: 是的



**我来也**

2019-02-09

👍 1

[D34打卡]

平常只用netstat 和 ifconfig ，前面的专栏里学了sar观测网络指标，今天又接触了两个类似的：ss和ip。

平常遇到的网络问题比较简单，先看能否正常连上，再看看并发连接数。有时忘记执行ulimit -n会导致默认账号的一个进程同时打开文件数只有1024。...

展开 ▼

**加盐铁论**

2019-02-25

👍

打卡，加油💪！

展开 ▼

**MJ**

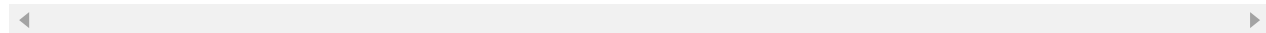
2019-02-25

👍

老师，带宽和吞吐量指标。区分上下行吗，？还是统计总量？

展开 ▼

作者回复: 嗯嗯，分为接收和发送

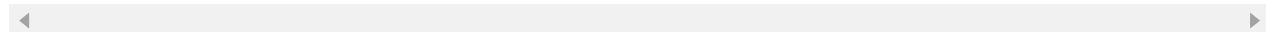
**Gaoyc**

2019-02-22

👍

通过ifconfig和ss看到的错误包或丢弃包等的一些错误是累加的嘛？是否可以清空这些错误包信息？

作者回复: 是的，都是累加值，所以不建议清空这些统计信息。并且，真正要清的化，也需要停止网卡并且卸载（rmmod）网卡内核模块，这在实际环境中通常是不允许的。

**--SNIPER**

2019-02-20

👍

老师好，netstat -anu 输出中：网卡收发队列时不时的会排队500，这种该如何再深入排查下是哪里的问题

作者回复: 注意区分下状态, Established 状态表示字节数, 500应该是正常的; Listening 状态的话, 可以去查查半连接



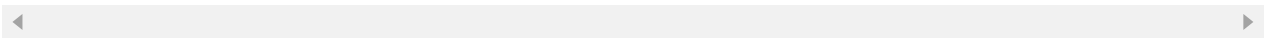
ninuxer
2019-02-15



打卡day36

去年之前喜欢用netstat, ifconfig, 去年年中的时候入坑ss, ip

作者回复: 嗯嗯



小美
2019-02-12



吞吐量, 表示没丢包时的最大数据传输速率

这个翻译有点问题?

Throughput is how much data actually does travel through the 'channel' successfully.

展开 ▾

作者回复: 嗯嗯, 有点别扭, 我来稍微调整下

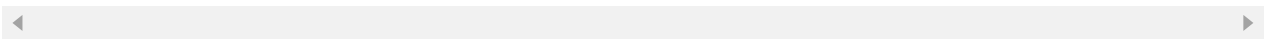


陈云卿
2019-02-10



老师, 有没有什么办法可以知道路由器的带宽是不是被打满了? 除了路由器的图形监控之外. 怎么从连接到路由器上的机器知道路由器的带宽情况?

作者回复: 监控就是最好的方法了。从其他机器上也可以Telnet到路由器上查看



Scott
2019-02-08



你好，我man一下netstat，看到Foreign Address的定义如下

Foreign Address

Address and port number of the remote end of the socket. Analogous to "Local Address."

...

展开 ▼

作者回复: 用了 -l 选项把其他非监听的连接都过滤掉了，去掉 -l 就可以了

