

linux下TC HTB流量控制Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_linux_E4_B8_8BTC_c103_645357.htm C规则涉及到队列(Queue) 分类器(Class) 过滤器 (Filter) ,filter划分的标志位可用U32或iptables的set-mark来实现) 一般是"控发"不控收 linux下有两块网卡,一个eth1是外网,另一块eth0是内网.在eth0上做HTB。

(注意: filter划分标志位可用u32打标功能或iptables的set-mark功能, 如果用iptables来打标记的话, 下行速LV在eth0处控制, 但打标应在进入eth0之前进行, 所以, “-i eth1”.例子: 主要命令就下面三句: 创建一个HTB的根 1.tc qdisc add dev eth0 root handle 1: htb default 20 创建一个HTB的类,流量的限制就是在这里限制的,并设置突发. 2.tc class add dev eth0 parent 1: classid 1:1 htb rate 200kbit(速率) ceil 200kbit burst 20k(突发流量) 创建一个过滤规则把要限制流量的数据过滤出来,并发给上面的类来限制速度 3.tc filter add dev eth0 parent 1: prio 1(优先级) protocol ip u32 match ip sport 80 0xffff flowid 1:1 说明: 让交互数据包保持较低的延迟时间, 并最先取得空闲带宽, 比如: ssh telnet dns quake3 irc ftp控制 smtp命令和带有SYN标记的数据包, 都应属于这一类。为了保证上行数据流不会伤害下行流, 还要把ACK数据包排在队列前面, 因为下行数据的ACK必须同上行流进行竞争。 TC IPTABLES HTB SFQ 1 tcp/ip 协议规定, 每个封包, 都需要有ACKNOWLEDGE讯息的回传, 也就是说, 传输的资料需要有一个收到资料的讯息回复, 才能决定后面的传输速度, 并决定是否重新传输遗失的资料, 上行的带宽一部分就是用来传输这些ACK资料的. 上行带宽点用大的时候, 就会影

响ACK资料的传送速度,并进而影响到下载速度,2 试验证明,当上传满载时,下载速度变为原来速度的40%,甚至更低,,因为上传文件(包括ftp上传,发邮件SMTP),如果较大,一个的通讯量令带宽超向包和,那么所有的数据包按照先进先出的原则进行排队和等待,这就可以解释为什么网内其中有人用ftp上传文件或发送大邮件的时候,整个网速变得很慢的原因. 解决速度之道: 1 为了解决这些速度问题,对经过线路的数据进行了有规则的分流.把本来在宽带上的瓶颈转移到我们的LINUX路由器上,可以把带宽控制的比我们购买的带宽小一点. 这样,我们就可以方便的使用tc技术对经过的数据进行分流与控制. 我们的想像就像马路上的车道一样,有高速道,还有小车道,大车道,需要高速的syn ack icmp ssh等走高速道,需要大量传输的ftp-data,smtp等走大车道,不能让它堵塞整条马路,各行其道. linux下的TC(traffic control)就有这样的作用,只要控制得当,一定会有明显的效果.tc 和iptables结合是最好的简单运用的结合方法. 我们设置过滤器以使用iptables对数据包进行分类,因为iptables更灵活,而且还可以为每个规则设置计数器,iptables用mangle链来mark数据包,告诉了内核,数据包会有一个特定的FWMARK标记值(handle x fw) 表明它应该送给那个类(classid x:x),而prio是优先值,表明那些重要数据应该优先通过那个通道,首先选择队列(选择htb),一般系统默认的是fifo的先进先出队列,就是说包是按照先来先处理的原则,如果有一个大的数据包在前面,那么后面的包只能等前面的发完后才能接着发了,这样就算后面既使是一个小小的ack包,也要等待了,这样上传就影响了下载,就算你有很大的下载带宽也无能为力. HTB(Hierarchical token bucket,分层的令牌桶),就像CBQ一样工作,但是并不靠计算闲

置时间来整形,它是一个分类的令牌桶过滤器.,它只有很少的参数. 结构简图: 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 www.100test.com