流量控制工具 TC 原理及应用

倪 飞

(重庆大学计算机学院 重庆 400030)

摘 要 随着多媒体通信在 Internet 中运用的日益广泛,越来越多的 Internet 应用都提出对流量控制的需求。本文介绍了 Linux 平台下功能非常强大的流量控制工具 TC,讨论了 TC 的重要组成部分队列规定、过滤器、类的相关细节,最后给出了一个运用 TC 工具实现流量控制的具体实例。

关键词 流量控制,TC,Linux,服务质量

1 引言

TC 是 Linux 下一个功能非常强大的流量控制工具,由 Alexey Kuznetsov 开发,从 Linux 2.2 版开始并入 Linux 内核。它把队列规定(Queuing discipline)、类(Class)、过滤器(Filter)结合在一起,其性能可以和专业的带宽管理系统相媲美。流量控制的根本目的就是保证网络的服务质量。因为 IP 协议族平等地对待任何业务,提供尽力而为型服务(best-effort service),在这种情况下对某些实时性要求比较的业务就很难保证其服务质量。引入流量控制概念,就是力图区分不同的用户或业务,根据其需求的不同分配不同的带宽。带宽不同,得到的服务质量自然也不同。如图 1 所示,TC 相当于给用户提供了一个进行流量控制的接口,通过这个接口就可以改变 Linux 内核对其接收到的数据包的处理方式。

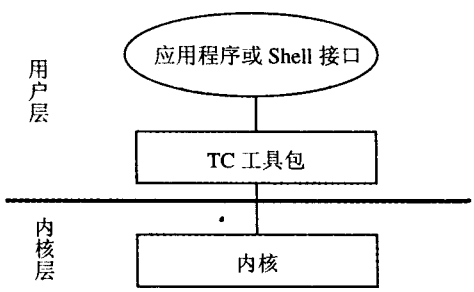

图 1

2 TC 的基本原理

TC 主要包括队列规定(Queuing discipline)、过滤器(Filter)、类(Class)这三个关键组件。队列规定是 TC 最重要的组成部分,实质上是一些算法,它规定了对进入队列的数据包的处理方式。过滤器(又称分类器)则根据一些参数或准则将进入队列的数据包分为不同的类。类由队列规定来管理,它和队列规定紧密联系在一起。不同的类对应着不同的队列规定,类本身不储存数据包,而是利用队列规定来管理其所拥有的数据。

如图 2 所示,当 Linux 内核接收到从用户级传来的数据包时,先根据过滤器(也称分类器)进行分类,每个过滤器都有相应的过滤条件,符合条件的数据包就被归为相应的类,而每一个类都对应一个相应的队列规定。值得一提的是,如果一个数据包对于所有的过滤条件都不满足,则该数据包会被归为缺省类,将会使用缺省的队列规定对其进行处理(缺省的队列规定一般是 pfifo_fast)。

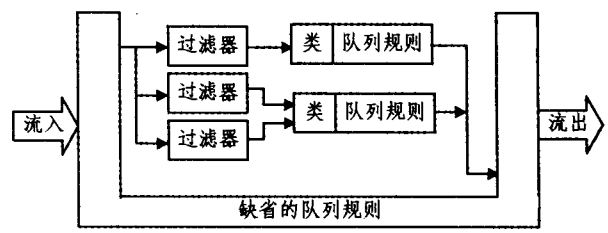

图 2

3 队列规定(Queuing discipline)

可以将队列规定视为设备的流量/数据包管理器。在队列规定内封装了其他两个主要 TC 组件(类和过滤器),它是流量控制(traffic control)的基础。无论何时,内核如果需要通过某个网络接口发送数据报,它都需要按照为这个接口配置的 QDISC(队列规定)把数据报加入队列。然后,内核按照规则取出数据报,把它们交给网络适配器。

在 Linux 中队列规定被分成两类:CLASSLESS QDISC(不可分类的 QDISC)和 CLASSFUL QDISC(可分类的 QDISC)。

3.1 不可分类的队列规定

(1) pfifo_fast:pfifo_fast 的特点就像它的名

字——先进先出(FIFO),也就是说没有任何数据包会被特殊对待。这个队列有3个所谓的"信道"。每一个信道都遵循FIFO(即先进先出)规则。只是在0,1,2这三个信道之间存在一定的优先级。如果在0信道有数据包等待发送,1信道的包就不会被处理,1信道和2信道之间的关系也是如此。

(2)SFQ(Stochastic Fairness Queueing,随机公平队列):SFQ是公平队列算法家族中的一个简单实现。它的精确性不如其它的方法,但是它在实现较高公平的同时,需要的计算量却很少。SFQ将流量被分到相当多数量的FIFO队列中,每个队列对应一个会话。数据按照简单轮转的方式发送,使得每个会话都按顺序得到发送机会。

(3)TBF(Token Bucket Filter,令牌桶过滤器):TBF是一个相对简单的队列规定。它只允许以不超过事先设定的速率到来的数据包通过,它的实现在于一个缓冲器(桶),这个桶不断地被一些叫做"令牌(Token)"的虚拟数据以特定速率(token rate)填充。最重要的桶参数就是它的大小,也就是桶能够存储令牌的数量。TBF的一个重要规则是,只有获得令牌的队列才能得到发送数据包的机会。

3.2 可分类的队列规定

(1)CBQ(Class Based Queueing,基于类别排队):CBQ具有丰富的连接共享类别结构,它既有限制(shaping)带宽的能力,也具有进行带宽优先级管理的能力。它通过计算连接的空闲时间来实现对带宽的限制的。

(2)HTB(Hierarchical Token Bucket,分层的令牌桶):HTB就像CBQ一样工作,但它并不靠计算空闲时间来整形。它实际上是一个分类的令牌桶过滤器。它的优点在于只有很少的参数,更易于配置。使用HTB可以很容易地保证每个类别的带宽,但它也允许特定的类可以短时内突破带宽上限,占用别的类的带宽。HTB可以通过TBF(Token Bucket Filter)实现带宽限制,也能够划分类别的优先级。

(3)PRIO:PRIO队列规定并不进行整形,它仅仅根据所配置的过滤器把流量进行进一步细分。可以认为PRIO队列规定是pfifo_fast的一种衍生物,两者之间的区别在于每个信道都是一个单独的类,而非简单的FIFO。当数据包进入PRIO队列规定后,将根据给定的过滤器设置选择一个类。缺省情况下有三个类,这些类仅包含纯FIFO队列规定,你可以把它们替换成任何需要的队列规定。使用PRIO队列规定可以很容易对流量进行优先级管理,只有属于高优先级类别的数据报全部发送完毕,才会发送属于低优先级类别的数据报。

4 类(Class)与过滤器(Filter)

类是与队列规定是紧密联系在一起的。正如上节所提,可分类的QDISC可以包含一些类。每个类都只有一个父类,而一个类可以有多个子类。此外,每个类都有一个叶子QDISC,默认情况下,这个叶子QDISC使用pfifo_fast的方式排队,也可以使用其它类型的QDISC代替这个默认的QDISC。而且,这个叶子QDISC也可以分类,不过每个子类只能有一个叶子QDISC。当一个数据报进入一个分类QDISC,它会被归入某个子类。可以使用tc过滤器(tc filter)、服务类型(Type of Service)等多种方式为数据报归类,如果数据报没有被成功归类,就会被排到这个类的叶子QDISC的队中。

过滤器根据数据报的某些特征来区分数据报,如数据报的源IP地址、目地IP地址,协议类型、TOS字节、网络接口,端口等。目前,LINUX可以使用的过滤器有:fwmark分类器,u32分类器,基于路由的分类器和RSVP分类器(分别用于IPV6、IPV4)等;其中,fwmark分类器允许我们使用Linux netfilter代码选择流量,而u32分类器允许选择基于ANY头的流量。过滤链表被保存在队列规定和类中,当一种队列规定的排队函数被激活,相应的过滤函数也被激活,用来对数据报进行分类。

5 一个具体的应用

先假设一个简单的环境,如图3所示。

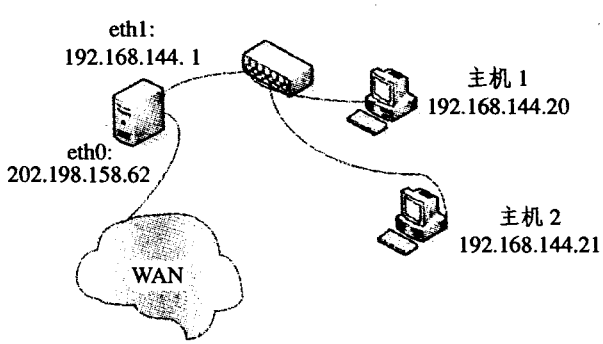

图3

在一个局域网中有三台主机,其中有一台主机作为网关具有两张网卡eth0和eth1,eth0具有合法的IP地址202.198.158.62,eth1分配私有网络地址192.168.144.1,其余两台主机也分配私有网络地址:192.168.144.20,192.168.144.21。

现在我们需要对流向主机1和主机2的流量进行如下控制:

(1)将发往主机 1 (192.168.144.20)的流量带宽控制在 8Mbit。

(2)将发往主机 2 (192.168.144.21)的流量带宽控制在 2Mbit。

5.1 建立队列

一般而言,针对一个网卡只需建立一个队列,我们对网卡 eth1 绑定 CBQ 队列:

tc qdisc add dev eth1 root handle 10:0 cbq bandwidth 10Mbit avpkt 1000

这句命令的意思是:将一个 cbq 队列绑定到网络物理设备 eth1 上,其编号为 10:0;网络物理设备 eth1 的实际带宽为 10 Mbit,包的平均大小为 1000 字节。

5.2 建立分类

针对一个队列需建立一个根分类,然后再在其上建立子分类。对于分类,按其分类的编号顺序发生作用,编号小的优先;一旦符合某个分类匹配规则,则通过该分类发送数据包。(在这个例子中,我们有一个根类 10:1 和两个子类 10:2 和 10:3)

(1)创建根分类 10:1

tc class add dev eth1 parent 10:0 classid 10:1 cbq bandwidth 10Mbit rate 10Mbit maxburst 20 allot 1514 prio 8 avpkt 1000 cell 8 weight 1Mbit

该分类的最大可用带宽为 10Mbit,实际分配的带宽为 10Mbit,可接收冲突的发送最长包数目为 20;最大传输单元加 MAC 头的大小为 1514 字节,优先级别为 8,包的平均大小为 1000 字节,包间隔发送单元的大小为 8 字节,相应于实际带宽的加权速率为 1Mbit。在这里有一个调节参数 weight,对带宽进行加权处理。

(2)创建分类 10:2,其父分类为 10:1

tc class add dev eth1 parent 10:1 classid 10:2 cbq bandwidth 10Mbit rate 8Mbit allot 1514 weight 800kbit prio 5 maxburst 20 avpkt 1000 bounded

该分类的最大可用带宽为 10Mbit,实际分配的带宽为 8Mbit,可接收冲突的发送最长包数目为 20 字节;最大传输单元加 MAC 头的大小为 1514 字节,包的平均大小为 1024 字节,包间隔发送单元的大小为 8 字节,相应于实际带宽的加权速率为 800Kbit。加上"bounded"参数,说明不应超过这一限制,否则这个 class 就要从别的 class 借带宽

(3)创建分类 10:3,其父分类为 10:1,

tc class add dev eth1 parent 10:1 classid 10:3 cbq bandwidth 10Mbit rate 2Mbit allot 1514 weight 200kbit prio 6 maxburst 20 avpkt 1000 bounded

该分类的最大可用带宽为 10Mbit,实际分配的带宽为 2Mbit,可接收冲突的发送最长包数目为 20 字节;最大传输单元加 MAC 头的大小为 1514 字节,包的平均大小为 1000 字节,相应于实际带宽的加权速率为 200kbit。"bounded"参数的作用与前面相同。

5.3 建立过滤器

过滤器主要用于分类。一般的作法是为根分类提供一个过滤器,然后为每个子分类提供路由映射。

(1)应用路由分类器建立 cbq 队列(即 10:0 所对应的队列)的根 tc filter add dev eth1 parent 10:0 protocol ip prio 100 route

(2)建立路由映射分类 10:2,10:3

tc filter add dev eth1 parent 10:0 protocol ip prio 100 u32 match ip dst 192.168.144.20 flowid 10:2

tc filter add dev eth1 parent 10:0 protocol ip prio 100 u32 match ip dst 192.168.144.21 flowid 10:3

5.4 建立路由

下面建立的路由必须前面所建立的路由映射一一对应。

(1)发往主机 192.168.144.20 的数据包通过分类 2(即前面建立的 10:2 类)转发 iproute add 192.168.144.20 dev eth1 via 192.168.144.1 realm 2。

(2)发往主机 192.168.144.21 的数据包通过分类 3(即前面建立的 10:3 类)转发 iproute add 192.168.144.21 dev eth1 via 192.168.144.1 realm 3。

结论 本文对 Linux 下强大的流量控制工具 TC 进行了分析,讨论了队列规定、过滤器、类的相关细节,在文章最后给出了一个实现流量控制的具体实例。实践证明通过编写 TC 脚本可以很方便、很灵活地实现对局域网的流量控制,改善网络的服务质量。

参 考 文 献

1 Hubert B, Gregorymaxzwell, Van Mook R. Linux Advanced Routing & Traffic Control HOWTO. http://lartc.org/lartc.pdf. 2003

2 Brown M A. Traffic Control HOWTO. http://tldp.org/HOWTO/Traffic-Control-HOWTO. 2003

3 Balliache L. Quality of Service. http://opalsoft.net/qos/QOS.htm. 2003

4 Blake S, Black D, Carlson M. An Architecture for Differentiated Services(RFC 2475). http://rfc.net/rfc2475.html. 1998