

Understanding Traffic Policing

Introduction

It is an increasingly common requirement for service providers to control the amount of data that their subscribers present to the network. Usually a subscriber has contracted for a given rate and the service provider must measure the data presented to ensure that, on one hand, the system is accepting the amount of data contracted for, and on the other hand, the user isn't trying to push a lot more data.

There are three different terms that are used to describe this process, *meter*, *police*, and *mark*. Although they are not exactly the same, over time these have come to be used almost interchangeably.



Policing is the term that describes the whole process—the process of monitoring network traffic for compliance with a traffic contract and taking steps to enforce that contract¹. A *policer* performs three primary functions: it *meters* each packet to determine whether it is in conformance with the service agreement; if it is, it *marks* the packet with its level of conformance (also called coloring); if it isn't, it *drops the packet*.

Policing can be done a number of different levels. It can be done at the port level (the UNI), or per Ethernet Service (the EVC), or for a particular class of service.

¹ Wikipedia - "Traffic policing"

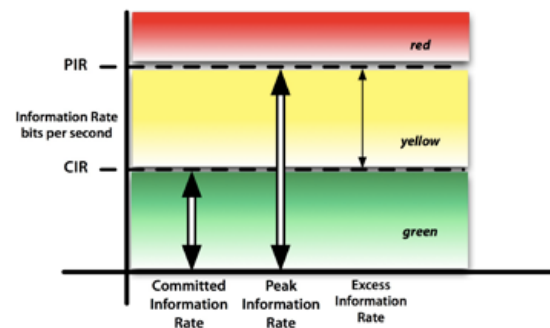
The BTI Systems packetVX Ethernet Switch and the BTI700-series Ethernet Access Devices², like many other Ethernet switches, can perform policing on incoming traffic flows.

The purpose of this whitepaper is to describe the parameters associated with policing, explain how to interpret them, how to choose the proper settings for them, and how to actually set them.

Two Rates

Simple rate limiting can be thought of as a *single rate policer* where a single rate value, for example 80Mbps, is specified. It is far more common these days to apply a *two rate policer* often called a *two-rate, three-color marker (TRTCM)*.

Not surprisingly, the two-rate, three-color marker is based on two information rate specifications as shown in the following diagram:



The most important rate is the *Committed Information Rate* or CIR. This is the data rate that the service provider is guaranteeing to the subscriber. A service level agreement will usually specify that all traffic presented at a rate of CIR or less will be delivered to its destination with high probability (usually above 99%).

² Most of the policing explanations in this paper apply generically to all devices that do packet policing, the BTI701 and BTI702 switches do not support the full range of capabilities described in this whitepaper.

Often the provider will allow the subscriber to present traffic at a rate higher than CIR and will attempt to deliver the *excess* traffic on a best-efforts basis. If the network is provisioned to provide enough transport capacity to carry the sum of all subscriber CIRs, then it is likely to be underutilized most of the time since it is unlikely that all subscribers will be presenting traffic at the CIR at the same time (this is the effect of statistical multiplexing). The service provider doesn't save any money running the network under-utilized, so it might as well allow the subscribers to use some of the excess bandwidth, either as a free benefit or for an additional fee. As long as the additional traffic can get shoved aside for committed traffic, everybody wins. This higher rate is called the *Peak Information Rate* or PIR.

There is a third rate noted in the diagram, the Excess Information Rate or EIR. This rate is just the difference between PIR and CIR. PIR has been commonly used for at least 10 years and is described in RFC2698³ in the context of IP traffic. When the MEF codified its recommendations for bandwidth metering in MEF5 (and then MEF10), it chose to describe the meter in terms of CIR and EIR rather than CIR and PIR. Obviously the difference is just one of arithmetic. We will use CIR and PIR in the remainder of this memo because PIR is a more generally understood term.

Three Colors

With a two-rate meter, each received packet can fall into one of three categories:

- Green—at or below CIR
- Yellow—Above CIR but at or below PIR
- Red—Above PIR

The common approach to handling this traffic is to forward green traffic—and try extra hard to deliver it since it is within the committed rate, mark yellow traffic—and forward it with best effort, and discard red traffic. Most systems allow these behaviors to be modified in some way—for example some systems might want to forward red traffic as if it were yellow (but maintain a separate counter for it).

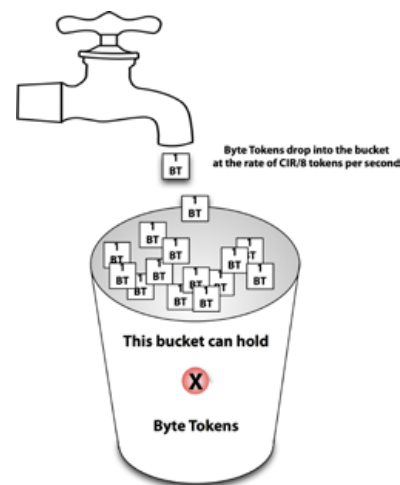
Token Buckets

Understanding how these meters are implemented (or could be implemented) can help in understanding exactly how they work and what they do.

Let's say that we want a CIR of 50Mbps. If we could just clock the Ethernet link down to that rate then we wouldn't need to do anything at all in the switch. The data could only come in at 50Mbps and we would be all done. Of course this would make it hard to have multiple services on the same Ethernet and it would make it hard to support a higher, best efforts rate (EIR/PIR). But we can't do it anyway, so it doesn't matter.

Even though we want to limit the flow to 50Mbps, when a packet arrives, it is arriving at the link speed (e.g. 1Gbps). Therefore, an average reception rate over some period of time must be computed as each packet arrives, and then the packet needs to be categorized based on whether the stream is at or below CIR, in which case the packet is considered green, or whether it is yellow, or red. This can be done using a mechanism called a token bucket.

Consider a bucket that can hold X Byte Tokens, each of which confers the right to receive one byte of data. Tokens are dropped into the bucket at the rate of $CIR/8$ tokens per second (CIR is specified in bits so we divide by 8). When the bucket is full no more tokens can go into the bucket—they just drop on the floor. The bucket starts out full.



Every time a packet arrives, we try to remove a Byte Token for each byte in the packet—a 1500 byte packet removes 1500 Byte Tokens. *If there aren't 1500 Byte Tokens in the bucket, then the average packet rate has exceeded CIR and the packet is not green.*

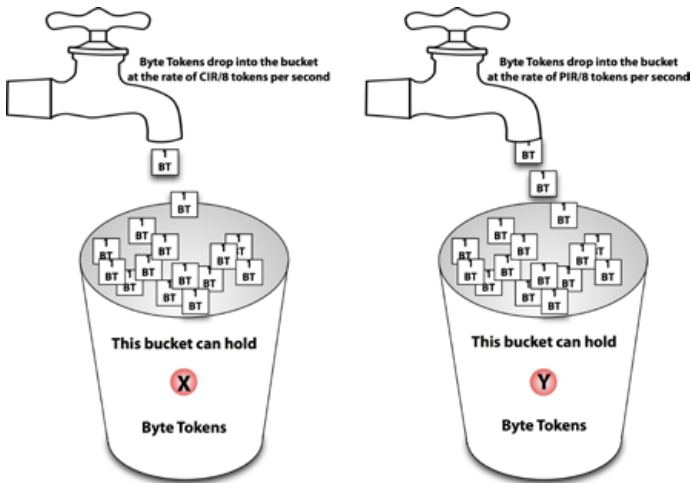
Now consider if there were actually two buckets as shown in the diagram below. The original one that can hold X Byte Tokens and is refilled at a rate of $CIR/8$ tokens per second, and a second one that can hold Y Byte Tokens and is refilled at a rate of $PIR/8$ tokens per second. Both buckets start out full.

When each packet arrives, we try to remove⁴ a token from **both** buckets for each byte in the packet. If there are enough tokens in bucket 1 (CIR) to cover the packet, then the packet is green. If there are not enough tokens in the CIR bucket, but there are enough tokens in bucket 2 (PIR) to cover the packet then the packet is yellow. If there aren't enough tokens in either bucket (independently) to cover the received packet, then it is red.

⁴ Tokens are only removed from a bucket if there are enough tokens in the bucket for the whole packet. Otherwise no tokens are removed from that bucket.

³ RFC2698 - A Two Rate Three Color Marker, September 1999, Heinanen, J. and Guerin R.

The nice thing about using the token bucket model is that it avoids quantization effects from averaging intervals. It provides a mechanism for implementing a continual **running average**. For example, for each packet received, before metering it, tokens can be added to the buckets representing the time duration since the last packet.



Picking the Values

If we use this approach we have four values that must be configured for each stream of traffic to be metered: CIR, PIR, X (size of the CIR bucket), and Y (size of the PIR bucket).

CIR is the easiest. This is just the committed bit rate that is being provided to the subscriber. Choosing CIR is a business decision, not a technical decision.

Choosing PIR is also a business decision, but the choice has technical implications. PIR allows access to unused bandwidth on both the access link and in the backbone. The first thing to consider is that as a link approaches its capacity, the average packet latency goes up. Having unused bandwidth has the benefit that it provides lower packet delay and lower packet delay variation. If the service provider is making latency guarantees as well as bandwidth guarantees, then this must be taken into account.

Further, if a large PIR is provided to one or more subscribers, those subscribers have the ability to hog the available bandwidth. So it might make sense to spread the bandwidth over the user base in a more controlled fashion (e.g. CIR+x%).

That leaves the choice of X and Y .

Assume that we set X to 500 Byte Tokens. The bucket starts full. The first packet to arrive is 1500 bytes long. Well, clearly it is not green as there aren't 1500 Byte Tokens in the bucket. In fact it can *never* be green. This yields the first rule, X (and Y) must be at least as big as the largest packet that can be received on the link, or otherwise there will never be a green packet. X and Y provide the ability to receive a short term burst of packets from the Ethernet. As discussed earlier, the packets will arrive at line rate regardless of what the committed rate is, so the system must be able to buffer at least one packet at line rate (and probably a small burst of a few packets). So X and Y represent a burst size, and are called the *Committed Burst Size* (CBS) and *Peak Burst Size* (PBS) respectively.

Another way to look at these values is that they represent how much received data the system is willing to buffer for the flow in the steady state. If the CBS is 1522 bytes (the maximum standard Ethernet packet including a C-VLAN tag), then the system is only willing to buffer one maximum-sized packet (or 23 minimum-sized packets). Once the CIR bucket is empty, any packets that arrive will be yellow, at best.

Let's plug in some numbers to understand the mechanism better. Assume that CIR is chosen to allow one thousand 1500-byte packets per second (about 12Mbps). CBS is set to 1500 and the bucket starts full. The first packet received empties the bucket but is marked green. The bucket starts filling up again. It takes 1 ms to fill completely. If another 1500-byte packet is received, say, 1/2 ms later, we can add 750 byte tokens before metering it, but it will not be green. If another 1500-byte packet is received at the 1ms mark (or beyond), it is green. So the system is willing to accept and buffer one 1500-byte green packet every millisecond.

Note that this doesn't guarantee that the system will only have one 1500-byte packet buffered. It is assumed that since the bandwidth is committed, that in normal, steady-state, operation, the first 1500-byte packet would be forwarded on before the next one is received. But things aren't always normal. There could be some short term congestion, re-routing, etc. going on so that the first packet hasn't yet been forwarded when the second one is received. This isn't a problem but it does mean that the actual buffering requirements could be more than the expected buffering requirements in these cases.

Now we can take into account PIR and PBS. Assume PIR is set to 24Mbps, (twice the CIR). If CBS and PBS are both 1500 then the system will accept a burst of one green 1500-byte packet each millisecond and a burst of two 1500-byte yellow packets each millisecond. The fourth packet to be received within a millisecond will be red.

CBS and PBS selection must provide a balance between the bursting capabilities needed by the user flow and the buffer availability in the switch. A larger burst size can result in greater TCP throughput since it allows a larger transmit window. The packetVX and BTI700 switches allow CBS and PBS to set in 4K increments.

A CBS setting of 8K should be sufficient for many services with a maximum packet size of 1522, although 16K or more might be appropriate for subscribers with very bursty/high throughput applications. If the maximum packet size is 9600 bytes then 12K to 20K should be fine for many services. Remember that most traffic flows have a mix of packet sizes and therefore the number of packets that fit in the buckets is greater. A 4K bucket can hold about 32 minimum-sized packets, for example. PBS can be set to the same value as CBS.

Setting the Values

On the packetVX, the policing values are defined as part of a *Bandwidth Profile* which can then be applied to an ESERVICE UNI as part of a Policy Map.

Note: In release 7.3.x and earlier the packetVX uses the keywords EIR and EBS to represent PIR and PBS. This will be addressed in a future release of the packetVX.

A bandwidth profile can be defined as follows:

```
> PROFILE BANDWIDTH name
> POLICE CIR value
> POLICE CBS value
> POLICE EIR value
> POLICE EBS value
> EXIT
```

For CIR and EIR, the value can be specified as xKbps, xMbps, or xGbps.

For CBS and EBS, the value is specified as the number of Kbytes.

For example:

```
> PROFILE BANDWIDTH frank
> POLICE CIR 12Mbps
> POLICE CBS 4
> POLICE EIR 24Mbps
> POLICE EBS 8
> EXIT
```

On the BTI700 switches, the policing parameters are specified using the METER command.

```
> METER <1-256> CIR <1-160000> CBS <1-10000>
  PBS <1-10000> PIR <1-160000>
```

The CIR and PIR specifications are in units of 4K bytes and the CBS and PBS specifications are in units of bytes. So, to specify a meter with the same values

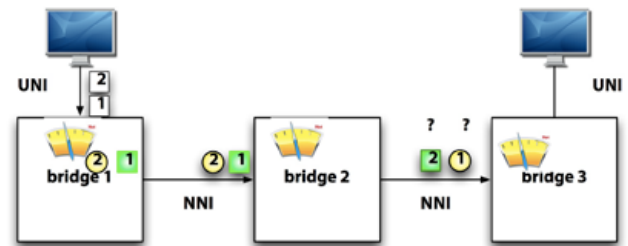
as the packetVX example above, the BTI 700 command would be:

```
> METER 1 CIR 188 CBS 4000 PBS 8000 PIR 375
```

A few other topics

Color-aware vs. Color-blind metering

Most policing is done in a color blind mode. In other words, it is done without any knowledge or interest about whether the packets were metered and marked at a previous step. Consider the following diagram:



If bridge 2 is metering traffic coming in from its NNI, then some of the traffic coming in was green (square) after ingress to bridge 1 and some of it was yellow (round). This doesn't matter if bridge 2 is doing color-blind metering. Each packet gets a new lease on life. A yellow packet could now be green and vice versa.

On the other hand, if the bridge 2 is doing *color-aware* metering, then a packet can never *get better* than it was. If it was green, it can be metered as green, yellow, or red, but if it was yellow, then it can only be yellow or red. The situation shown in the diagram could not occur. Packet 1 could go from green to yellow after passing through bridge 2's meter, but packet 2 could not go from yellow to green. The *color marking* at the previous hop(s) can be encoded in the priority-code-point (PCP) portion and DEI field of the VLAN tag or in the DSCP field.

It is not common to police packets at a point beyond the ingress UNI unless the packet is crossing service provider domains—for example, if bridge 1 was in one service provider's network and bridge 2 was in another's. In this case the receiving service provider might police the total traffic flow at the NNI port level to ensure that the other service provider is not pumping more traffic into the network than agreed upon.

Within a single provider's domain, however, this usually doesn't make sense. For a subscriber presenting a packet at a UNI, the packet has a fixed cost/value, and it will either be delivered or not. However, for the service provider, the value of each packet increases (in a sense) for each hop that it traverses. The

packet uses backbone bandwidth as it travels from hop to hop which increases the service providers investment in the packet and makes it less and less desirable to discard the packet. If the packet makes it all the way to its destination switch, dropping the packet makes almost no sense at all. If 20% of the traffic is TCP traffic⁵ then a dropped packet has a one in five chance of being re-transmitted, using the bandwidth all over again.

Single-Rate Three-Color Meter

In addition to the Two-Rate Three-Color meter most systems implement a single-rate three-color meter (SRTCM). The operation is similar to the TRTCM in that there are two token buckets, but the buckets are filled at the same rate, CIR. The size of the buckets are CBS and EBS (excess burst size).

With the SRTCM, the service provider is not allowing the subscriber to send excess traffic at a greater rate than the committed rate, but will allow a burst of traffic to exceed CBS. If CBS and EBS are both 4K, for example, an incoming packet will remove tokens from the CIR/CBS bucket if there are enough tokens available. In this case the packet will be green. If there are not enough tokens available in that bucket then they will be removed from the CIR/EBS bucket and the packet will be marked yellow. If there are not enough tokens in either bucket, the packet is red.

One subtle difference between the TRTCM and the SRTCM is that in the TRTCM tokens for a packet are taken out of both buckets (if they are available) and in SRTCM they are only taken out of the second bucket if there are not enough in the first bucket. This is because in the TRTCM the second bucket is a peak bucket and is getting filled faster than the first bucket, but in the SRTCM it is an excess bucket and is getting filled at the same rate as the first bucket.

⁵ This used to be a lot more, possibly over 80%, but with increasing bandwidth being used for voice and video it has been dropping.

Egress Policing

Policing is primarily a process that is done on packets coming into the system on a port. There are a few cases where it makes sense to police packets on egress. The primary case is when there are more than two end-points to the flow—a LAN service rather than a LINE service. In a point-to-point service the egress on one side can't be more than the ingress at the other, but in a multi-point service it can be.

If an Ethernet Service has three end-points and each one has a 100Mbps CIR then it is possible for two of the end-points to send at up to the full CIR to the third at the same time. If the service provider intends the service to be symmetric (i.e. CIR in and CIR out) then policing the traffic at the egress can accomplish this.