Creating a Bandwidth Management Strategy

Why Implement Bandwidth Management?

In a world where clients, servers, and data all come in different types, treating all network connections as a single traffic flow does not give administrators the flexibility to ensure that business-critical applications perform as required. Without Bandwidth Management, network administrators may be forced to block bandwidth-intensive applications like streaming audio and video. However, with Bandwidth Management, administrators can instead allow controlled access to these less-than-critical but still useful tools. Bandwidth Management allows administrators to control network traffic flow so that the appropriate users, groups, or applications receive the proper network resources. These include:

- Preventing resource-greedy applications – such as peer-to-peer (P2P) and streaming video – from impacting other, more important programs, such as e-mail and client-server applications.
- Allowing administrators to provide for applications that require a per-session amount of bandwidth. For instance, multimedia applications may require a certain amount of bandwidth in order to enable all their features.
- Enabling administrators to balance important bandwidth-intensive protocols and applications, such as HTTP and Instant Messaging (IM).

Understanding Bandwidth Management

Types of Traffic Supported

Bandwidth Management can be used to control most of the traffic that is handled by Blue Coat ProxySG appliances. The network administrator can specify custom classifications for each type of traffic, which are in turn used to manage bandwidth. These can then be organized into a bandwidth hierarchy to implement the required bandwidth management strategy and policies. Managing bandwidth for the following traffic types is supported:

- HTTP/HTTPS
- FTP
- Streaming
- P2P
- IM
- SOCKS
- TCP Tunnel
- Bridging
Bandwidth Classes

A bandwidth class is the basic unit of bandwidth management used to group traffic into distinct partitions to be managed. It provides the ability to set up minimum bandwidth guarantees, maximum bandwidth limits and priorities for a given type of traffic. Bandwidth classes are created from the Management Console or Command Line Interface (CLI) and can be organized into a hierarchy (a tree) in order to enable bandwidth sharing. Two bandwidth classes with the same parent can share their excess/unused bandwidth with each other.

Implementing Bandwidth Classes Using Policy

Simply creating a bandwidth class does not complete the task of managing bandwidth using that class. The Blue Coat ProxySG must be instructed as to which traffic belongs to which of the many bandwidth classes configured on the box. This is done by creating policy. Using the VPM or writing Content Policy Language (CPL), the underlying language of VPM, the administrator can classify traffic into a bandwidth class. The policy triggers used to perform this classification depends on the nature of bandwidth policy being implemented.

Classifying Traffic

Application-Based Classification

Setting bandwidth classes based on the application type (which maps to a service type in the Blue Coat ProxySG) is the simplest form of bandwidth management. The Blue Coat ProxySG can be configured with a set of bandwidth classes, one for each service type. The mapping from service type to bandwidth class can be modified via the Management Console or CLI.

Directional-Based Classification

Traffic for any application can be classified based on direction of flow of data using the limit_bandwidth() CPL property.

- client inbound [CPL: limit_bandwidth.client.inbound()]
- client outbound [CPL: limit_bandwidth.client.outbound()]
- server inbound [CPL: limit_bandwidth.server.inbound()]
- server outbound [CPL: limit_bandwidth.server.outbound()]

For any application, the administrator can write policy to classify each of these into a separate bandwidth class. For the predefined classes, policy automatically classifies all four of these into the same default class. This may not be sufficient for certain types of traffic. For example, all HTTP traffic will get classified into the default http class. If one has to limit HTTP WAN bandwidth utilization as typically is the case, setting limits on the default http class will not do what is intended as it will limit client side traffic (inbound traffic, including cache-hits) to this configured limit also.

Policy-Based Classification

Traffic can be classified using most of the policy triggers available in CPL. This type of classification can be used for fine grain control of your bandwidth. For example, instead of classifying all P2P traffic into a single P2P class, one may want to classify based on the P2P protocol being used or based on which subnet the client belongs to. The full power of CPL policy language can be applied to create very granular classification of traffic.
How Bandwidth Management Works

The Blue Coat ProxySG bandwidth management solution uses queue-based packet management to properly prioritize or limit traffic flows. Before classification, all traffic (including non-TCP traffic) is part of the default bandwidth class. Determinations of which specific bandwidth class each transaction belongs to is policy driven and takes effect as soon as the policy rule can commit. Thus a user-based prioritization will not become effective until user authentication completes. Packets in the traffic flow before the bandwidth management rule commits will be handled as part of the default bandwidth class associated with the interface/bridge in the system. The ethernet headers per packet count against the bandwidth class to which the transaction has been applied, but activities such as ARP and DNS necessary to service a transaction are not included.

With the Blue Coat Proxy SG’s method of bandwidth management, TCP will behave exactly as it would on a network which had a bottleneck link. For “server inbound” bandwidth management on the ProxySG, it would appear as if the bottleneck link were on the last hop just before the ProxySG. For other devices doing bandwidth management using a similar technique, the bottleneck might appear somewhere else, but the effects would be similar.

Creating a Bandwidth Management Strategy

In general, there are four reasons to manage bandwidth:

- Protect performance of mission-critical application (e.g. SAP).
- Prevent bandwidth-greedy applications from impacting others (e.g. P2P).
- Provision bandwidth for applications that need a per-session amount of bandwidth (Streaming).
- Balance necessary and important (but bandwidth-greedy) applications (e.g. HTTP, IM).

In order to fulfill your requirements, the following strategy should be used as a guideline:

1. Categorize all applications on your network into four types:
   - Applications that are mission-critical (e.g. SAP, Oracle). These are critical to organizational success.
   - Applications that are impacting performance (e.g. P2P). These typically have large flows and are bandwidth hungry, expanding to consume all available bandwidth.
   - Applications that are low-bandwidth but sensitive to latency (e.g. telnet).
   - Applications that are sensitive to jitter (e.g. Streaming). These require smooth, consistent delivery for their operation.

2. Create bandwidth classes for each application on these lists using the Management Console or CLI. Note that each application’s traffic can map to multiple bandwidth classes depending on the granularity of control you need. Typically you at least need a server-side class and a client side class for each application. You might also want separate classes for inbound and outbound traffic on client and server-side.

3. Write Policy to perform the correct classification for your application to the user-defined bandwidth classes. At this point, if bandwidth management is enabled, you can let the ProxySG run with no bandwidth limits configured and collect bandwidth management statistics. This will help you determine the pattern of application bandwidth usage on you network. This information will help you set the correct limits for the bandwidth classes for each of your applications.
After evaluating your statistics, you can set up bandwidth limits on the mission-critical bandwidth classes to match your requirements. For mission-critical applications, you will typically set up minimum bandwidth. This ensures that your mission-critical applications get the required share of bandwidth to perform adequately. If these mission-critical applications are also bandwidth-greedy (e.g., HTTP), then you might also want to set up maximum bandwidth limits so that they don’t completely overwhelm the link thus leaving no bandwidth for all other applications.

Set up bandwidth limits for your unimportant bandwidth classes to match your requirements. For these applications, you will typically set up maximum bandwidth limits. This ensures that your unimportant traffic is restricted to its configured share of bandwidth.

Set up bandwidth limits for your latency-sensitive traffic. This is accomplished by creating a two level class hierarchy with a common parent class for all the classes in your lists and setting its maximum bandwidth to less than link bandwidth. Now using priority, you can give the latency-sensitive classes a higher priority than other classes. This ensures that if there is bandwidth available, the latency-sensitive traffic has first priority.

Set up bandwidth limits for your jitter-sensitive traffic. These are mostly streaming applications. The Blue Coat ProxySG already has the ability to do admission control for streaming traffic and this mechanism is preferred for managing bandwidth for this type of traffic. The admission control mechanism makes the end-user experience significantly better for streaming applications by avoiding thinning for admitted streams. If you prefer to allow every client in regardless of the effects of thinning, you can set a maximum bandwidth limit for these classes to restrict their bandwidth usage.

Group all the above classes into hierarchies if you have a need to share excess/unused bandwidth between groups of classes. For example, if you want to set up an overall limit for all HTTP server inbound traffic but also want to specify limits for individual departments, you might want to create a two-level hierarchy with an HTTP server inbound class and a subclass for each of the departments. This will allow you to set bandwidth limits for each department as well as for overall traffic.

### Additional Bandwidth Management Information

#### Access Log Uploads

The Blue Coat ProxySG can be configured to limit the bandwidth used by access log uploads. Using access log CLI commands, or using the Management Console, you can specify a bandwidth class to use to manage bandwidth for uploads. Typically you only need to set the maximum bandwidth limit on this class. If you are upgrading from 3.x, the upgrade procedure will automatically create bandwidth classes for each of the access log facilities configured at the time of upgrade.

#### Service Info

The Blue Coat ProxySG can be configured to limit the bandwidth used by service info uploads. The Blue Coat ProxySG has a default service-info class that limits the amount of bandwidth used for uploads; the administrator can set the maximum bandwidth limit on this class for their specific bandwidth needs.