EXTRAHOP WHITE PAPER

Ten Best Practices for Optimizing ADC Deployments

Abstract

Application Delivery Controllers (ADCs) are critical control points in the application environment. This paper examines the complexities around deploying ADCs and focuses on some best practices for deploying ADCs such as the BIG-IP product family from F5 Networks.

Table of Contents

© 2009 ExtraHop Networks, Inc. All rights reserved.

Introduction

Load balancers were created to distribute web requests across a pool of servers in order to improve scalability and availability. Early load balancers were simple catch-and-release systems that buffered packets and then released them while applying smart network address translation. Over time, load balancers evolved into Application Delivery Controllers (ADCs) as they took on additional features, such as TCP optimization, SSL termination, and HTTP caching. Today, modern ADCs are critical components of the enterprise network infrastructure.

However, with these advanced capabilities comes greater complexity and management challenges. This white paper explores ten best practices for optimizing deployments of ADCs, such as the BIG-IP product family from F5 Networks.

ADC Deployment Challenges

Complex configuration settings – ADCs provide intelligent traffic management and high availability for mission-critical applications. In order to achieve these goals, ADCs such as BIG-IP come with an array of configuration parameters to enable customers to match their ADC deployment to the environment. However, when configured incorrectly, the ADC degrades performance rather than improves it. Unfortunately, the optimal configuration settings are highly dependent on your specific network topology and application mix. The default settings provided by the vendor are only a starting point.

Straddling the network and application layers – ADCs are in-line network devices that blur the lines between the network and application layers. They perform complex content inspection and transformations across all layers of the protocol stack. They rewrite IP addresses, change HTTP headers, split and aggregate transactions across different TCP connections, and even serve cached content directly. Problems with any of these functions can impact both the network and application teams. In such scenarios, isolating the root cause of the problem can be quite difficult.

Dearth of health and performance metrics – ADCs excel at intelligent, high-speed traffic management. However, they expose only limited metrics for performance tuning, troubleshooting, and capacity planning. This limitation is a common theme for most in-line network-infrastructure devices, which focus on moving large volumes of data at very high speeds rather than providing visibility into that data. In general, network devices do not always provide the best estimation of their own health, especially when their health is poor and you most want to know about it.

Interactions with other infrastructure components – ADCs sit at critical points within the enterprise network, often interacting with hundreds or thousands of systems, applications, and databases as well as other networking devices. With so many moving parts, the IT infrastructure is approaching a tipping point of complexity. Performance degradation and failures are increasingly more common and more difficult to isolate.

© 2009 ExtraHop Networks, Inc. All rights reserved.

Best Practices for Deploying ADCs

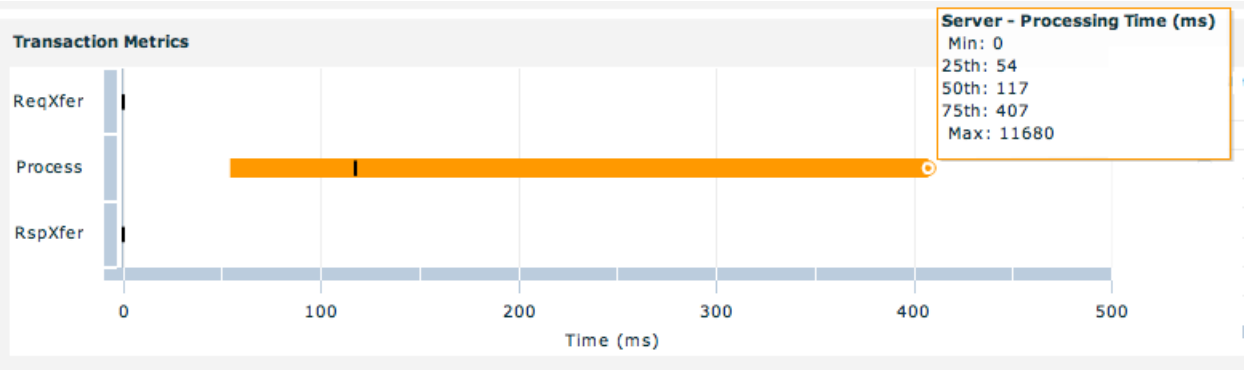
#1: Establish baselines for performance

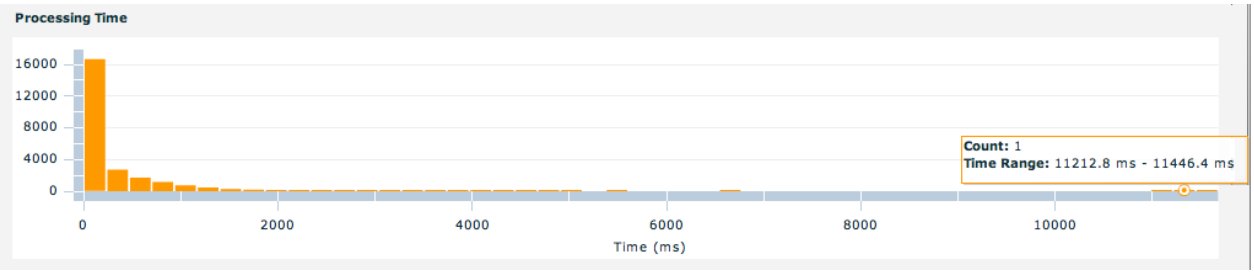
Problem: ADCs often are blamed for performance problems. They are complex and largely opaque devices that process business-critical network traffic. ADCs can degrade performance in certain situations. Other times, they are blamed unfairly.

Impact: Performance problems are almost always difficult to isolate, resulting in a lot of wasted time and effort on the part of the network-engineering team.

Implementation: Comparing performance of the virtual server, or vip, with that of the back-end pool members will help locate the cause of performance problems. This comparison can be achieved through the use of an Application Delivery Assurance system.

In the example below, the ExtraHop system shows a clear breakdown of the total transaction time in terms of request transfer time, server-processing time, and response transfer time. The median server-processing time is 117ms. This value should be the same on the virtual server and the back-end pool members. Looking at the timing distribution chart for server-processing time, it is clear that most transactions are processed quickly, but there are a few outliers at over 11 seconds. If these outliers are present on the back-end pool members as well as the front-end virtual server, then the slow transactions most likely are caused by an application issue rather than a problem with the ADC.





© 2009 ExtraHop Networks, Inc. All rights reserved.

#2: Analyze transactions in real time

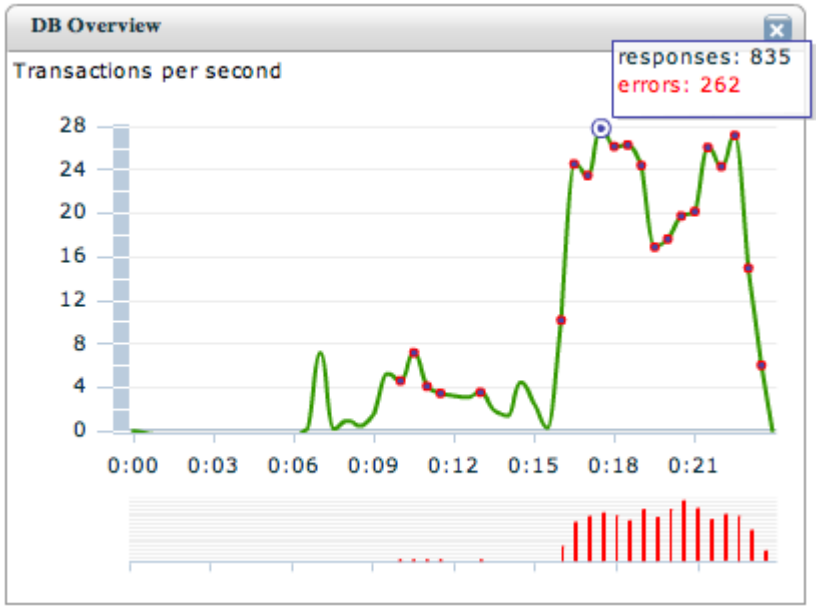
Problem: ADCs use periodic service checks to determine server health. These checks have several shortcomings:

- First, due to the periodic nature of the checks, there is an inherent undersampling problem. If a server fails intermittently, then the service checks most likely will miss it.
- Second, synthetic transactions do not always fail in the same manner as real-world ones. This issue is especially true for Internet-facing services.
- Finally, service checks are good for detecting hard failures but they are bad at detecting degraded services. If transactions take longer and longer to complete but not so long that the service check gives up, then the server still is considered healthy.

Impact: Ineffective service checks can cause ADCs to direct requests to failing servers, negatively impacting user experience.

Implementation: Real-time transaction analysis using an Application Delivery Assurance system provides a proactive, early-warning system, enabling you to address problems before they become disasters.

As an example, consider the following overview chart for database transactions on a network segment monitored by the ExtraHop system. Transactions began failing at about 12:15am. Not all transactions are failing, so periodic service checks most likely would succeed without detecting the problem. Real-time transaction analysis combined with trend-based alerts can notify the proper IT personnel when application errors occur or when transaction-processing time is anomalously high.



© 2009 ExtraHop Networks, Inc. All rights reserved.

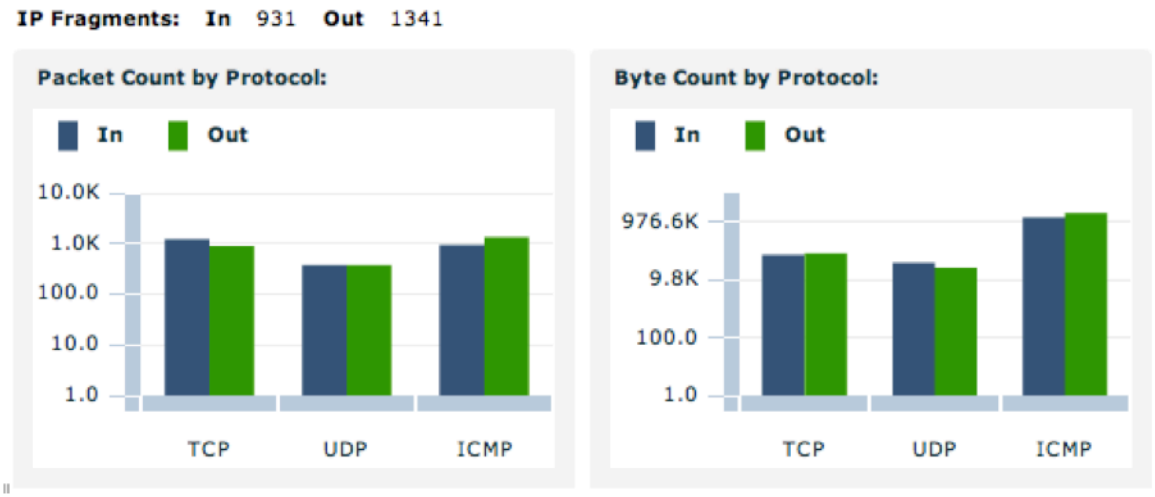
#3: Detect and avoid IP fragmentation

Problem: Anyone who has used a VPN knows that mismatched MTUs cause IP fragmentation and that fragmentation degrades performance. There are more packets flying around the network, and the fragments have to go through special reassembly processing within the IP stack. With reliable protocols, if a single fragment is lost, the whole datagram must be retransmitted.

Impact: IP fragmentation often degrades performance. TCP is engineered to avoid IP fragmentation through a process called path-MTU discovery. Unfortunately, path-MTU discovery rarely works over the Internet these days. Many firewalls are configured to filter out the ICMP fragmentation needed messages. Some firewalls and ADCs are simply unable to forward these messages across a NAT. BIG-IP is one of the few ADCs that handle this situation properly.

Implementation: Use an Application Delivery Assurance system to detect IP fragments or excessive ICMP path-MTU discovery messages on critical links. If you observe large numbers of fragments or excessive ICMP fragmentation needed messages on your network, you should take steps to address the problem.

The ExtraHop system tracks L3 metrics such as IP fragments as well as individual ICMP messages in real time.



#4: Ensure SNAT pools are sufficiently large

Problem: ADCs commonly are configured to SNAT (Secure Network Address Translation) to the backend servers. Due to the 16-bit port numbers, TCP and UDP have a limit of 64K simultaneous connections for each pair of IP addresses to a single destination port.

Impact: If the connection rate is too high and the number of unique IP addresses is too low, then new connections might fail intermittently. Most TCP stacks will wait at least three seconds before attempting to reconnect. When this problem occurs, it is very difficult to track down. Some clients will take 3, 6, 18 seconds or even longer to connect.

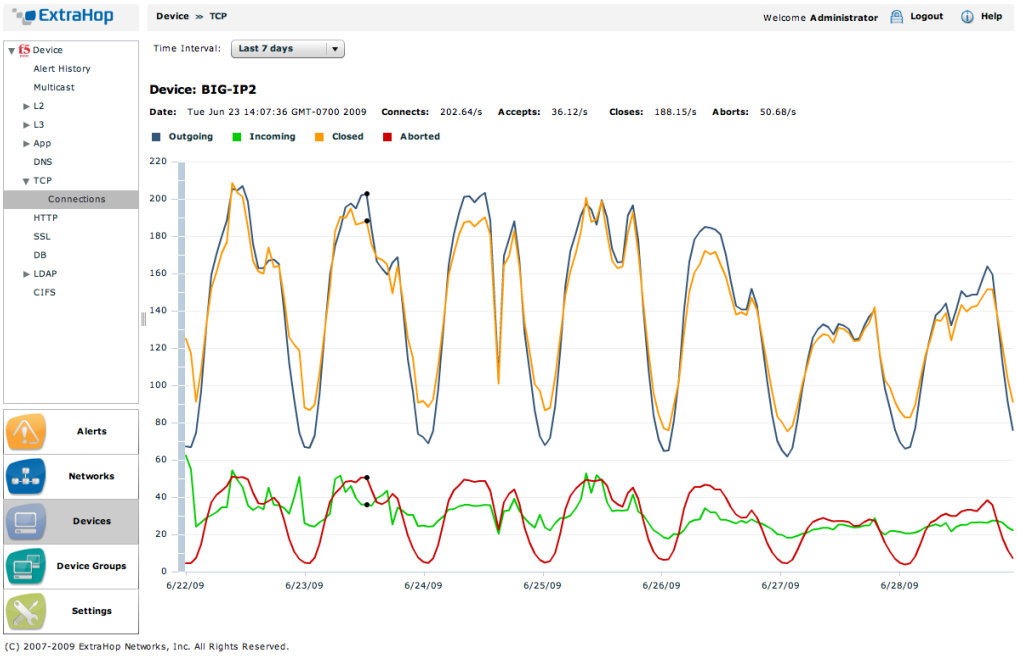
Implementation: When using the SNAT feature, you should make sure that there is a sufficiently large pool of unique IP addresses. On BIG-IP, this is called a SNAT pool.

© 2009 ExtraHop Networks, Inc. All rights reserved.

However, when using SNAT auto-map, the self IPs of the BIG-IP implicitly make up the SNAT pool.

When sizing your SNAT pools, a good rule of thumb for short-lived connections is to measure the total new connections per second to each backend server and then divide by the number of unique IPs. If the value exceeds 1000 or so, then the SNAT pool should be larger. While the ADC does provide some of these metrics, it does not account for internal connections such as service checks and it does not provide automatic alerts for IP address exhaustion.

An Application Delivery Assurance system can provide detailed information for existing BIG-IP deployments to help make this determination. In the chart below, the server is accepting 36 new connections per second. Connections are closed at roughly the same rate indicating that they are short lived. This value is well below 1000, so even a single IP address in the SNAT pool is sufficient.



#5: Provision load-balancing pools appropriately

Problem: One of the primary functions of an ADC is to distribute load across a pool of servers for scalability and availability. But how many servers do you need? Answering this question is an exercise in provisioning and capacity planning. Unfortunately, ADCs generally do not directly provide metrics to help with this exercise.

Impact: If the load-balancing pool is too small, then server load will be too high during peak traffic. There is a tipping point for many applications; once it is reached, the application just falls over, possibly resulting in disruption to business-critical operations.

Implementation: When server-processing time trends upward, it often is an indication of excessive load. Real-time analysis of server-processing time using an Application Delivery Assurance system can provide an early warning for insufficient capacity. Historical trends can help determine if there is sufficient headroom, both during peak times and during

© 2009 ExtraHop Networks, Inc. All rights reserved.

maintenance windows when a portion of the servers are removed from the primary load-balancing pool.

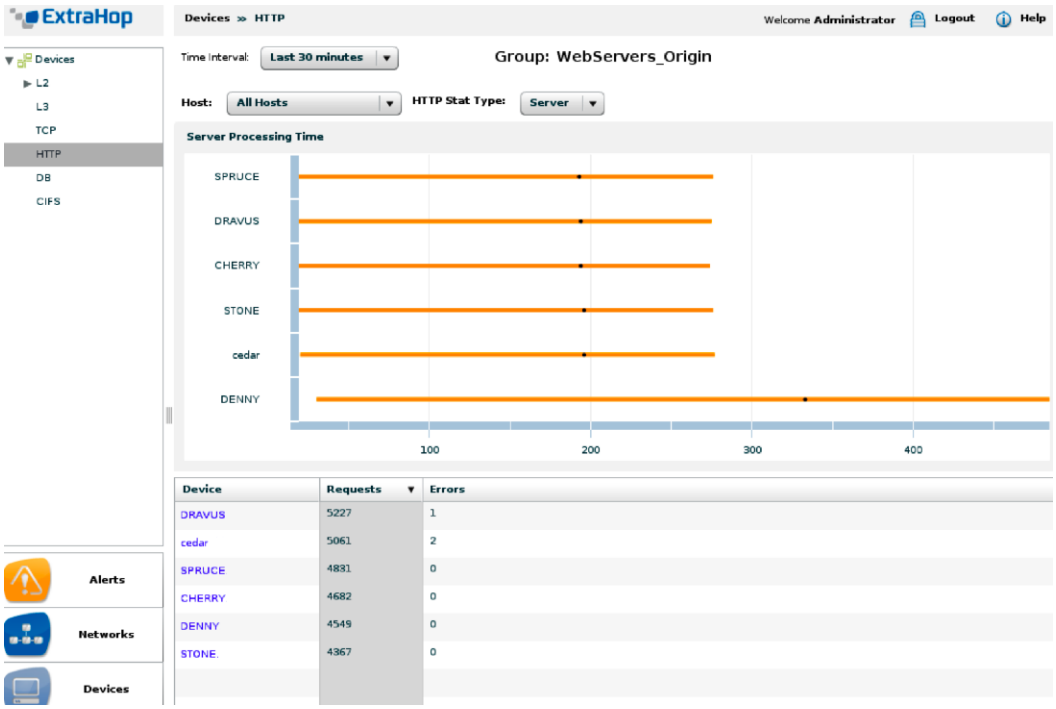
#6: Use proper load-balancing methods and ratios

Problem: Many load-balancing pools are comprised of heterogeneous hardware. Some servers are older and slower while others are extremely fast.

Impact: While ADCs like BIG-IP support a number of different load-balancing methods, the round-robin method is usually the default. For a heterogeneous pool with differing performance characteristics, round-robin or ratio-LB methods will result in too many connections being directed to the slow servers.

Implementation: To ensure optimal user experience and application response time, compare the performance of the servers within each LB pool. Identify the slow-performing servers and adjust the pool-member ratios as needed. On BIG-IP, consider using Fastest as the LB method; this method will direct traffic to the server with the fewest outstanding requests.

The ExtraHop Application Delivery Assurance system presents a side-by-side comparison of server-processing time for each server in a load-balancing pool. In this example, one server is slower than the others, and the busiest server has received nearly 20% more requests than the least busy server. The LB method or the ratio of the slow pool member should be adjusted.



© 2009 ExtraHop Networks, Inc. All rights reserved.

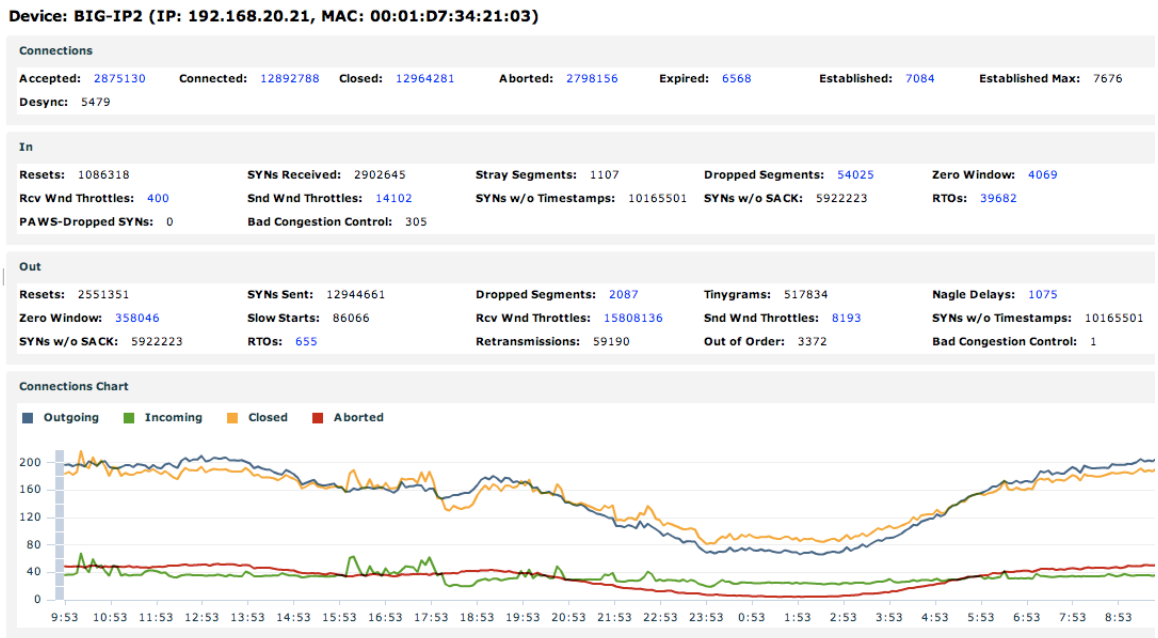
#7: Choose between FastL4, performance-HTTP, and full-proxy modes

Problem: BIG-IP has several different modes depending on which profiles are assigned to the virtual server. FastL4 and performance-HTTP modes offload the operational complexity of TCP to the endpoints. These modes require fewer resources, like processor and memory, on the BIG-IP itself. Full-proxy mode uses two complete TCP stacks, one for the client side and one for the server side. Other ADCs on the market have similar performance and full-proxy modes.

Impact: In certain situations, choosing the wrong mode can impact user experience negatively.

Implementation: The best way to determine if your ADC is operating in the correct mode for your specific network topology and application mix is to analyze the TCP connections running through the ADC using an Application Delivery Assurance system. Every connection should be analyzed as problems tend to be intermittent. Performance modes sometimes will experience problems with congestion control, excessive slow starts, and PAWS drops.

The diagram below displays the results of sophisticated TCP analysis provided by the ExtraHop system. In this example, the bad congestion control, slow starts, and PAWS drops metrics are acceptably low compared to the total number of connections. This information indicates that the ADC is running in the correct mode.



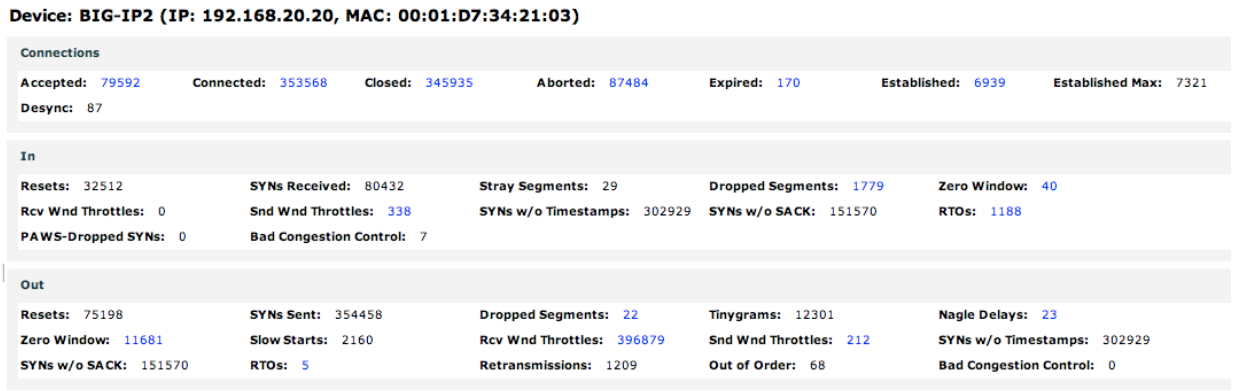
#8: Optimize TCP settings

Problem: When running in full-proxy mode, ADCs expose a number of TCP-related configuration settings. On BIG-IP, these settings are found in the TCP profile. ADC vendors generally try to include reasonable default values for these settings based on best-effort guesses. In the case of F5, BIG-IP ships with separate TCP LAN and TCP WAN profiles for the user to select accordingly. Unfortunately, the optimal settings are highly dependent on the specific network topology and application mix, and there are no indicators or metrics to help set or adjust them.

© 2009 ExtraHop Networks, Inc. All rights reserved.

Impact: When configured incorrectly, the ADC can degrade performance rather than improve it. In particular, Nagle's algorithm, proxy buffer thresholds, and ACK-on-push are three settings in the TCP profile that are difficult to set correctly.

Implementation: Incorrect or suboptimal settings can be identified by sophisticated TCP analysis using an Application Delivery Assurance system. In the example below, the ExtraHop system shows relatively high numbers of receive-window throttles as well as zero-window advertisements. Each zero-window advertisement can cause up to a 5-second stall in the flow. These values indicate that the ADC is pushing back and throttling the peer, and the ADC essentially is communicating that it cannot keep up. On BIG-IP and similar systems, this situation can indicate that the proxy buffer high threshold should be increased.



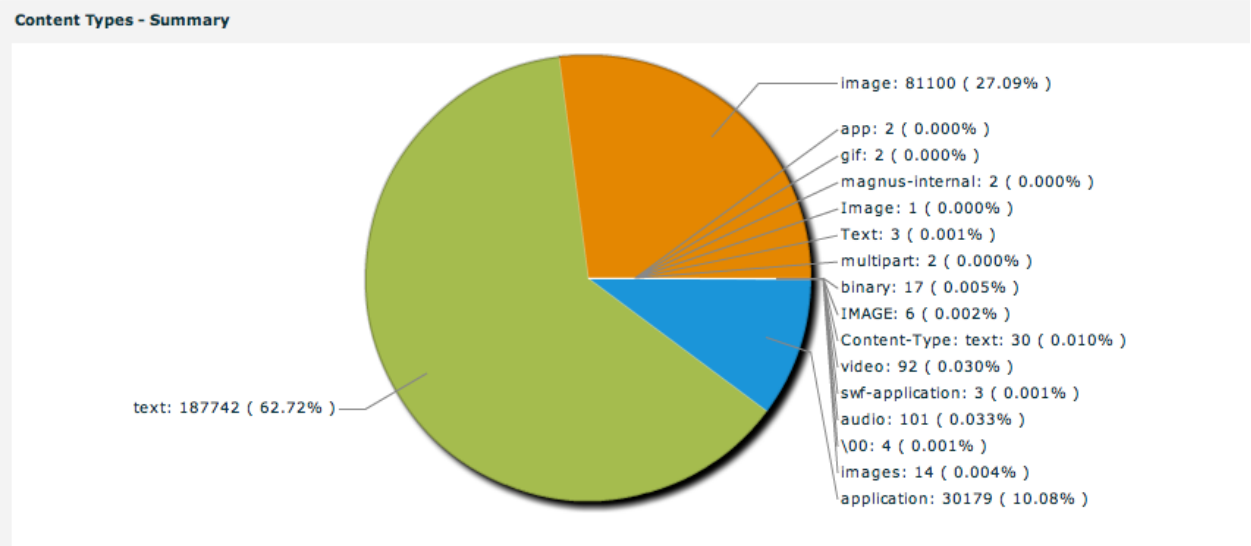
#9: Evaluate HTTP-caching policies

Problem: Modern ADCs support HTTP content caching, but often it is unclear when this feature should be enabled.

Impact: The BIG-IP Fast Cache and Web Accelerator features can improve user experience significantly. However, unnecessary or indiscriminate caching can cause unintended problems with some applications.

Implementation: Frequent requests for static, cacheable content are good candidates for HTTP content caching. As shown in the diagram below, the ExtraHop system monitors all HTTP requests in real time and provides reports on the content types as well as the individual URIs and the number of times they are requested.

© 2009 ExtraHop Networks, Inc. All rights reserved.

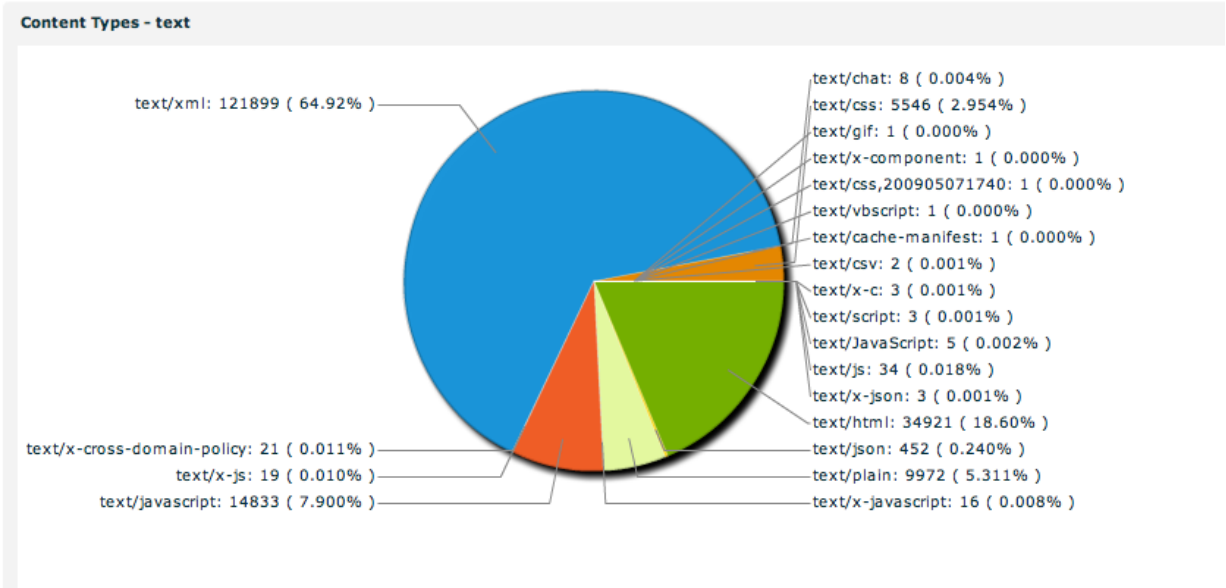


#10: Compress HTTP when appropriate

Problem: Modern ADCs support dynamic HTTP content compression. Like the HTTP caching features, often it is unclear when this feature should be enabled.

Impact: When used effectively, HTTP compression can reduce bandwidth and improve user experience. However, in some situations, unnecessary compression can cause additional overhead with little or no added benefit. Additionally, compressing some content, like JavaScript, is known to cause problems with certain browsers. By default, BIG-IP includes exceptions for many of these cases.

Implementation: Frequent requests for documents with content-type text are good candidates for HTTP compression. Once again, the content type reports provided by the ExtraHop system help to identify these candidates as shown in the diagram below.



© 2009 ExtraHop Networks, Inc. All rights reserved.

It is important to note that there is little need to compress documents on the LAN. In fact, the additional overhead for compression and then the time spent uncompressing the content on the client could be a net loss in terms of performance.

The ExtraHop system tracks the origin of each request, so it is easy to determine if content is traversing the WAN. The ExtraHop system also tracks the round-trip time for the TCP connections so that special attention can be paid to the high-latency connections.

Summary

Application Delivery Controllers are mission-critical and complex network devices. Best practices for ADC deployments must be followed to achieve optimal application performance and user experience. In many instances, an Application Delivery Assurance system, which measures how well applications are delivered over the complex network infrastructure, can help to implement these best practices and maximize your ADC investment.

About ExtraHop Networks

ExtraHop Networks was founded in early 2007 by Jesse Rothstein and Raja Mukerji, engineering veterans from F5 Networks and architects of the BIG-IP v9 product. The company's pioneering Application Delivery Assurance system is the industry's first completely passive network appliance that provides application-level visibility with zero agents, configuration, or overhead. The privately held company is headquartered in Seattle, Washington and funded in part by the Madrona Venture Group.

To learn more about ExtraHop and our innovative Application Delivery Assurance system, visit us at www.extrahop.com.

© 2009 ExtraHop Networks, Inc. All rights reserved. ExtraHop and the ExtraHop logo are registered trademarks of ExtraHop Networks, Inc.