

# Akamai Data Reference

This document is provided for reference purposes only. The specifics of your contract with Akamai Technologies supersede the information contained within the *Akamai Data Reference*. For billing information specific to your contract, please contact your Akamai representative.

April 2011

## **Akamai Technologies, Inc**

Akamai Customer Care: **1-877-425-2832** or, for routine requests, email **ccare@akamai.com**

The EdgeControl Management Center, for customers and resellers: **<http://control.akamai.com>**

US Headquarters  
8 Cambridge Center  
Cambridge, MA 02142

Tel: 617.444.3000  
Fax: 617.444.3001

US Toll free 877.4AKAMAI (877.425.2624)

For a list of offices around the world, see:  
**<http://www.akamai.com/en/html/about/locations.html>**

## **Akamai Data Reference**

Copyright © 2006-2011 Akamai Technologies, Inc. All Rights Reserved.

Reproduction in whole or in part in any form or medium without express written permission is prohibited. Akamai, the Akamai wave logo, and the names of Akamai services referenced herein are trademarks of Akamai Technologies, Inc. Other trademarks contained herein are the property of their respective owners and are not used to imply endorsement of Akamai or its services. While every precaution has been taken in the preparation of this document, Akamai Technologies, Inc. assumes no responsibility for errors, omissions, or for damages resulting from the use of the information herein. The information in these documents is believed to be accurate as of the date of this publication but is subject to change without notice. The information in this document is subject to the confidentiality provisions of the Terms & Conditions governing your use of Akamai services.

Apple and QuickTime are trademarks of Apple Inc., registered in the U.S. and other countries.

All other product and service names mentioned herein are the trademarks of their respective owners.

# Contents

## **PREFACE • 7**

Overview of Akamai Data Reference .....	7
Audience .....	7
Document Organization .....	7
Related Resources .....	8
Documentation Feedback .....	9

## **CHAPTER 1. DATA COLLECTION AND METRICS • 11**

Data Collection Systems .....	11
Real-Time Feeds .....	12
Log-Based Data Collection .....	12
Log Delivery Service .....	12

## **CHAPTER 2. CONTRACT USAGE REPORTS • 13**

Contract Usage Overview .....	13
Statistics and Definitions .....	14
95/5 .....	14
Page Views .....	15

## **CHAPTER 3. LOG DELIVERY SERVICE • 17**

How Log Delivery Works .....	17
About Akamai Logs .....	17

## **CHAPTER 4. ACCESSING DATA • 19**

EdgeControl Management Center .....	19
HTML Format .....	19
CSV Format .....	19
Recurring Email Reports .....	19
EdgeControl Web Services .....	20
EdgeControl MIB .....	20
Log Delivery .....	20

## **CHAPTER 5. REPORTING DATA OVERVIEW • 21**

Data sources & Integrity .....	21
Log-Based Data .....	21
Real-Time Feeds .....	22
Traffic Types .....	23
Edge, Midgress, and Origin Traffic .....	23
Request and Response Traffic .....	24
Traffic Data .....	25
Hits .....	25
Volume MB - Object Bytes and Overhead Bytes .....	26
Concurrent Streams .....	29
Requests .....	30
Page Views .....	30
URL Data .....	30

50-Hit Requirement .....	30
HTTP URL Hit Limit .....	31
Filters .....	31
ESI Fragments .....	31
Aggregation .....	31
Flash URLs .....	32
URL Data in HTML format .....	32
URL Data in CSV format .....	32
Visitors Data .....	32
Unique Visitors .....	32
Visitor Data Prerequisite .....	33
Visitor Data Granularity .....	33
‘Other USA’ Data in US States List .....	33
Data Granularity .....	33
Timezones .....	33
Important Considerations for Configurable Timezones .....	33
Data Latency .....	35
Customized Reports .....	35
Date Ranges .....	35
Multiple CP Codes .....	35
Recurring Email Reports .....	35
Recurring report details .....	36
Supported Email Clients .....	37
Suspended Report Delivery .....	37

## **GLOSSARY • 39**

## **APPENDIX A. DETAILED OVERHEAD CALCULATIONS • 43**

Packets .....	43
HTTP .....	44
Constants and Equations .....	44
Overhead Calculations .....	47
Streaming .....	48
Constants and Equations .....	48
Total Bytes Calculations .....	53

## **APPENDIX B. COMMON PITFALLS IN REPORT COMPARISONS • 55**

Successful and Error Transactions in HTTP Reports .....	55
LDS Log Data Different from Traffic Reports .....	55
‘Data not Final’ Message in Contract Usage Reports .....	55

## **APPENDIX C. RESPONSE CODES • 57**

000 Client-Side Abort .....	57
100 Range – Informational Status Codes .....	57
200 Range – Successful Status Codes .....	57
300 Range – Redirection Status Codes .....	58
400 Range - Client Errors Status Codes .....	58
500 Range - Server Error Status Codes .....	59
600 Range - Invalid Headers .....	59

**APPENDIX D. GEOGRAPHIC REGIONS • 61**

**APPENDIX E. INTERNET 500 RETAIL CATEGORIES • 63**

Retail Categories .....63



# Preface

## In This Preface:



- ◆ Overview of Akamai Data Reference • 7
- ◆ Document Organization • 7
- ◆ Related Resources • 8
- ◆ Documentation Feedback • 9

## Overview of Akamai Data Reference

This Data Reference provides an overview of the metrics, statistics, and data collection processes used by Akamai reports, including Traffic, Visitor, and URL Reports. Contract Usage reports, which provide a preview or invoice charges, and Log Delivery Service (LDS), which provides Akamai server logs for HTTP, Streaming, and FTP traffic, are briefly described here.

For more information about your contract usage or invoice, please contact your Akamai representative.

### Audience

This document is intended for users of Reports, Contract Usage, and Log Delivery. Readers should be familiar with the basic mechanism of Web and Streaming content delivery.

## Document Organization

This document is organized as follows:

Chapter	Description
Chapter 1	<i>Data Collection and Metrics</i> gives an overview of the systems used to retrieve data and the metrics used to talk about it.
Chapter 2	<i>Accessing Data</i> describes the different mechanisms available to you for retrieving data.
Chapter 3	<i>Log Delivery Service</i> gives an overview of Akamai's log delivery and log data.

Chapter	Description
Chapter 4	<i>Contract Usage Reports</i> gives an overview of the purpose of contract usage reports and general billing invoice components. This general information is provided for informational purposes only; for information specific to your contract and invoices, please contact your Akamai representative.
Chapter 5	<i>Reporting Data Overview</i> provides a general look at the data provided in reports, including traffic types, data granularity and latency, date ranges, and CP codes.
Glossary	<i>Glossary</i> provides definitions for useful industry and Akamai terms.
Appendix A	<i>Appendix A: Detailed Overhead Calculations</i> describes the calculation of protocol overhead components.
Appendix B	<i>Appendix B: Common Pitfalls</i> highlights the most common errors in comparing reports that might calculate data differently.
Appendix C	<i>Appendix C: Response Codes</i> provides a complete list of standard response codes.
Appendix D	<i>Appendix D: Geographic Regions</i> provides a complete list of regions that can appear in a report.
Appendix E	<i>Internet 500 Retail Categories</i> provides the 2006 list of retail categories used by the Competitive Benchmark - Retail report now available in beta for Site Accelerator.

## Related Resources

Documentation specific to each Akamai service is available on EdgeControl at [Support > Documentation](#). Documentation for Log Delivery Service is also available there.

The online help in the Traffic, Visitor, and URL reports provides data descriptions for the report you are viewing.



## Documentation Feedback

*documentation@akamai.com*


If you have a suggestion for making this document more useful, please let us know by sending a message to [documentation@akamai.com](mailto:documentation@akamai.com). Your comments will be reviewed and taken into consideration for the next release of this document.

If you have feedback regarding Traffic, Visitor, or URL reports, please use the link at the top right of the EdgeControl screen and fill out the form.

If you have a technical problem or question that needs immediate attention, please contact Akamai Customer Care by opening a ticket online using the Issue tracking System available on EdgeControl at Support > Open/View Support Cases.



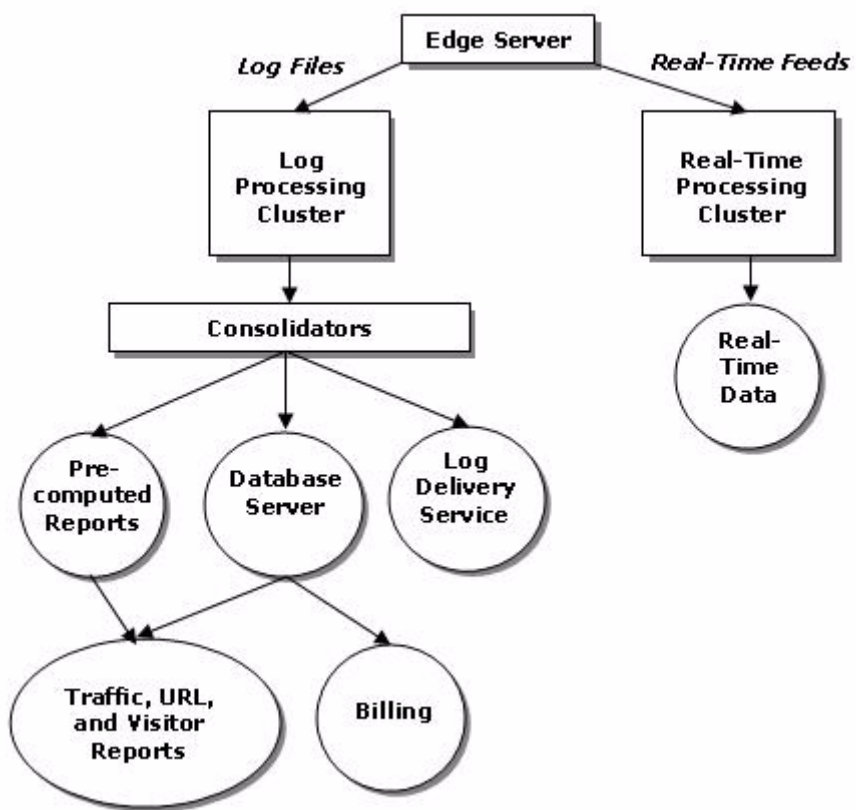
# Chapter: 1 Data Collection and Metrics

**In This Chapter:**  ◆ Data Collection Systems • 11  
◆ Log Delivery Service • 12

The chapter describes the systems Akamai uses to collect network data and how that data is made available to you.

## Data Collection Systems

Akamai has implemented two independent data collection systems: one based on log collection and one based on real-time network data feeds:



## Real-Time Feeds

The two data collection systems serve different purposes. The real-time system is designed to deliver a snapshot of current system state. Each edge server periodically publishes an update of recent activity to the Real-Time Processing Cluster, where the information is aggregated and made available to be queried. If transient Internet connectivity problems prevent contact, the real-time system will do without data from that particular edge server.

## Log-Based Data Collection

In contrast to the real-time feeds, the log-based data collection system drives the historical reports, and is optimized for data completeness. Edge servers send logs to the collection servers every hour (or more frequently, in some cases) and will continue to try to deliver the log until it is five days old, at which point it is discarded. That is why the log-based system is able to retrieve more data than the real-time system, and why the overall traffic level is sometimes higher in reports when compared to real-time data. The “tenacity” of the store-and-forward based log delivery system in retrieving a log accounts for the effect where data is seen to “trickle” into its reports over the course of a few days.

## Log Delivery Service


The Akamai Log Delivery Service (LDS) delivers edge server logs for HTTP, FTP (Net Storage), and Streaming traffic using a variety of formats and transmission options. Akamai provides LDS logs to enable analysis beyond that offered in the standard Reports, which is intended as a general-purpose reporting utility. Such analysis could include deep clickstream reports or proprietary aggregations based on URL query strings with other information private to the customer.

LDS logs contain transactions between Akamai edge servers and end users only, so aggregate statistics from these logs are best compared to Edge Egress Hits, Mbps, and MB from the Traffic reports, with the exception that the LDS log lines only contain the size of an object served. They do not contain any of the overheads described earlier. Because LDS logs contain only edge transactions, these logs can be compared easily to the logs produced by a content provider’s own HTTP and media servers pre-Akamaization.

Please visit [Support > Documentation > Log Delivery Service](#) for more information and sample log files.

..

# Chapter: 2      Contract Usage Reports

- In This Chapter:**  ◆ Contract Usage Overview • 13  
◆ Statistics and Definitions • 14

Akamai's Contract Usage reports focus on providing a preview of billing metrics on a month-to-date basis. This document provides general information about contract usage reports for information purposes only. For billing information specific to your contract, please contact your Akamai representative.

## Contract Usage Overview

Akamai's Contract Usage reports focus on providing a preview of 95/5 and month-to-date metrics throughout a billing month by computing month-to-date values for these metrics every night. These nightly month-to-date values are graphed as a time series so that content providers can extract usage trends. Of course, nightly 95/5 statistics are not necessarily good predictors of a month-end 95/5 if traffic patterns change significantly during the month.

The Contract Usage reports can also include superbursting computations. While traffic reports are CP code-based, Contract Usage reports are contract-based and take into account the correct CP code list for a contract, the contract start dates, any mid-month service upgrades, and service bundles. At the end of every month, the Contract Usage reports will include the exact measures that will appear on the Akamai invoices.

**NOTE:** Please note that the data displayed in the contract usage report is actual usage. The data will be rounded up to the appropriate increment for invoicing purposes.

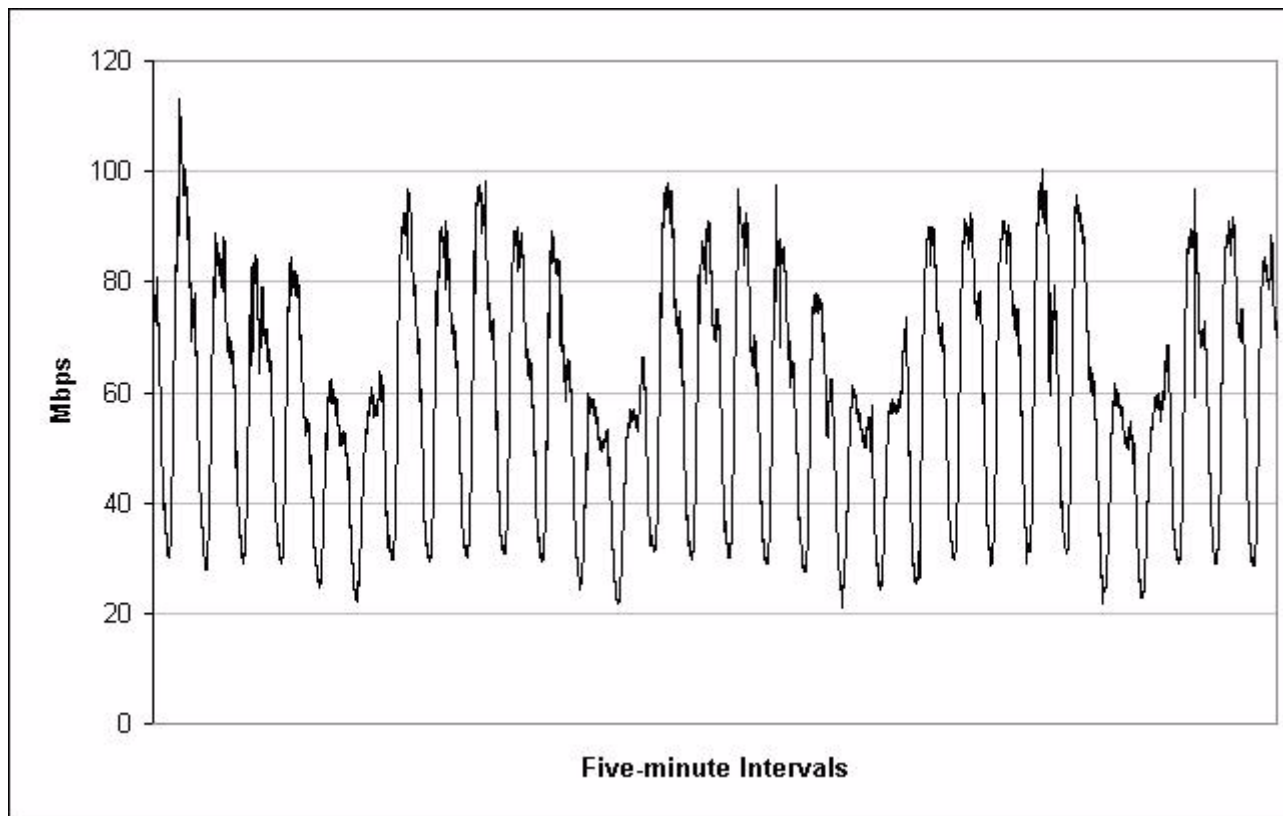
## Statistics and Definitions

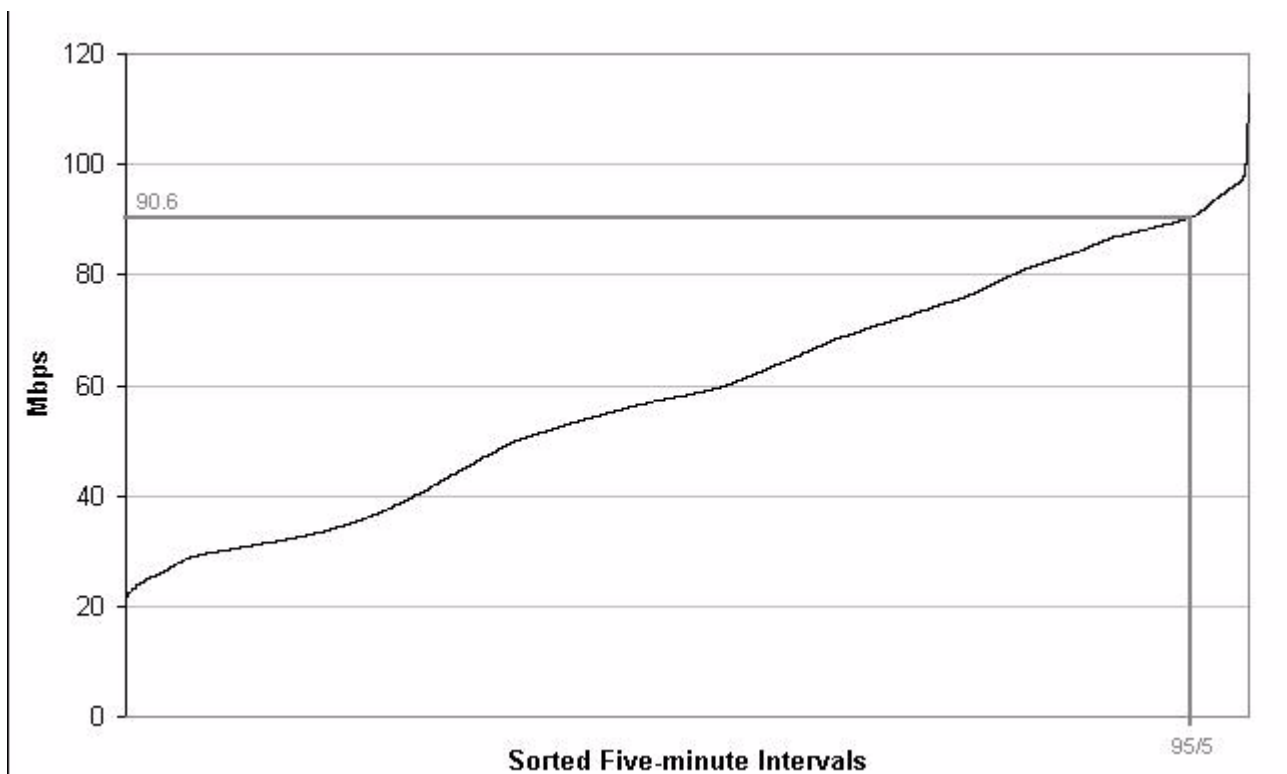
The metrics discussed in the previous section are collected to form time series of customer traffic. The reporting tools display these series graphically or summarize them using a variety of statistics. Some statistics merit additional explanation, including “95/5”, and minimum and maximum concurrent streams.

**95/5** The 95/5 statistic is used to determine a “near maximum” value for a time series, and it is commonly used in the networking industry to determine traffic rates. The “95” represents the “95th percentile” largest value of a time series, while the “5” indicates that the time series has data points every 5 minutes. Suppose you have an Edge Egress Mbps time series with data points for every five minutes during a 30-day month. This time series would consist of:

$30 \text{ days} \times 24 \text{ hours/day} \times 60 \text{ minutes/hour} \times 1 \text{ sample/5 minutes} = 8640 \text{ samples}$

If the samples were sorted from least to greatest, the 95th percentile sample would be sample 8208, corresponding to  $0.95 \times 8640$ . That sample would be the 95/5 statistic for the time series. The following two figures illustrate the procedure:





One of the interesting properties of the 95/5 statistic is that even a non-trivial amount of traffic can yield a statistic value of zero. For example, suppose that a content provider serves a steady 100 Mbps of traffic for exactly the first 24 hours (288 5-minute intervals) of a 30-day month and then serves 0 Mbps for the rest of the month. After sorting, only intervals 8353 and above would contain 100 Mbps of traffic. Interval 8208, the 95th percentile interval, would indicate 0 Mbps, so the 95/5 for the month for this traffic distribution would be 0 Mbps.

Other statistics can be described in the same way as the 95/5. A “98/5” statistic would represent the 98th percentile of a five minute time series. Similarly, an “80/60” statistic would be the 80th percentile of a time series with data points spread every sixty minutes.

Akamai generally uses the 95/5 statistic to calculate a monthly traffic number for Mbps usage. This statistic is also used to calculate a monthly storage consumption number for Akamai NetStorage as follows: individual 95/5 statistics are computed for each replication location, and the greatest of these statistics is used as the Net Storage consumption value for the month.


## Page Views

A page view is defined as the delivery of a file by Akamai that has a text/html content type but excludes redirects (HTTP response code 301/302) and File Not Found error page (HTTP response code 404). Akamai aggregates the number of text/html files delivered for a Web site each month.





# Chapter: 3      Log Delivery Service

**In This Chapter:**  ♦ How Log Delivery Works • 17  
♦ About Akamai Logs • 17

The Akamai Log Delivery Service delivers edge server logs for HTTP and Streaming traffic using a variety of formats and transmission options. Akamai provides logs to enable analysis beyond that offered in standard Reports. This analysis could include deep clickstream reports or proprietary aggregations based on URL query strings.

## How Log Delivery Works

Akamai's infrastructure is constantly gathering the log entries from the thousands of Akamai servers around the world. Log Delivery Service (LDS) creates a copy of these logs, separates your logs from other customer logs, and then delivers your logs based on a predetermined schedule. Most of the log files will be delivered within a 24-hour period.

 **Note:** There is no SLA for Log Delivery Service.

You must be a customer of Log Delivery Service to configure the service and begin receiving logs from that point forward. Logs are not available retroactively, and logs for other customers who do not subscribe to LDS are not retrievable. In addition, you must turn logging on when you configure your Akamai service.

## About Akamai Logs

LDS logs contain transactions between Akamai edge servers and end users only, so aggregate statistics from these logs are best compared to Edge Egress Hits, Mbps, and MB from the Traffic reports, with the exception that the LDS log lines only contain the size of an object served. They do not contain any of the overhead bytes. Because LDS logs contain only edge transactions, these logs can be compared easily to the logs produced by your own HTTP and media servers pre-Akamaization.

Please visit [Support > Documentation > Log Delivery Service](#) for more information and sample log files.



# Chapter: 4      Accessing Data

## In This Chapter:



- ◆ EdgeControl Management Center • 19
- ◆ EdgeControl Web Services • 20
- ◆ EdgeControl MIB • 20
- ◆ Log Delivery • 20

There are several ways for you to access data regarding the performance of your content or application and your contract usage.

## EdgeControl Management Center

The EdgeControl Management Center traffic, visitor, URL, and contract usage reports are available to you in two formats, HTML and CSV.

### HTML Format

The HTML format provides graphical and tabular representation of your data, with key data points called out. This format is available for traffic reports, visitor reports, URL reports, as well as contract usage reports.

### CSV Format

The CSV format provides report data in comma-separated value format. This format is available for traffic reports, visitor reports, URL reports, as well as contract usage reports.

Virtually all data, with the exception of FTP data and unique visitor data, has five-minute timestamps. Note that if there is no data for a particular timestamp, the row is not written to the CSV report.

Most CSV volume fields use the MB metric showing up to six decimal places, so rounding should be negligible or non-existent.

### Recurring Email Reports

You can configure reports to be automatically sent to you on a recurring basis, in either HTML or CSV format. You can receive reports on a daily, weekly, or monthly basis. Reports are only sent when data is 99% complete. See “Complete Data” on page 21 for more information.

## EdgeControl Web Services

EdgeControl Web Services allow you to access many EdgeControl features using the industry standard Simple Object Access Protocol (SOAP) interface instead of a browser. These options include provisioning, real-time and historical traffic, and contract reporting features of EdgeControl.

Please see the EdgeControl Web Services Developer's Guide located on EdgeControl at [Support > Documentation > EdgeControl Web Services](#) for more information.

## EdgeControl MIB

The EdgeControl MIB is designed to provide real-time HTTP and streaming traffic statistics, active EdgeControl alerts, and EdgeControl events directly to an end user's Enterprise Network Management System (ENMS). You can download the SNMP MIB application [here](#) and install it on a Windows® or Linux node of your network. When activated, the agent retrieves your real-time traffic data, alerts, and events information from Akamai Web Services servers and stores them in a local MIB. Once your ENMS console is aware of this new agent's MIB, it can then display the current values of specific MIB variables.

Please visit [Support > Documentation > EdgeControl MIB](#) for related documentation and downloads.

## Log Delivery

The Akamai Log Delivery Service provides you with the server logs from the various Akamai services that you are using, and is available for the following services:

- HTTP Content Delivery
- Streaming (QuickTime Streaming, Real Media Streaming, Windows Media Streaming, and Streaming for Flash)
- Net Storage (FTP)

Once you have a contract for Log Delivery Service, you can turn on logging and configure the delivery on EdgeControl. Please visit [Support > Documentation > Log Delivery Service](#) for related documentation and sample log files.

# Chapter: 5      Reporting Data Overview

## In This Chapter:



- ◆ Data sources & Integrity • 21
- ◆ Traffic Types • 23
- ◆ Traffic Data • 25
- ◆ URL Data • 30
- ◆ Visitors Data • 32
- ◆ Data Granularity • 33
- ◆ Timezones • 33
- ◆ Data Latency • 35
- ◆ Customized Reports • 35
- ◆ Recurring Email Reports • 35

This chapter provides information about reporting data in general, including how it is collected, types of data included in reports, and specific ways you can manipulate that data for your needs.

## Data sources & Integrity

### Log-Based Data

In contrast to the real-time feeds, the log-based data collection system drives the historical data, and is optimized for data completeness over response time. Edge servers send logs to the collection servers every hour (or more frequently, in some cases) and will continue to try to deliver the log until it is five days old, at which point it is discarded. That is why the log-based system is able to retrieve more data than the real-time system, and why the overall traffic level is sometimes higher in reports when compared to real-time data. The “tenacity” of the log-based system in retrieving a log accounts for the effect where data is seen to “trickle” into its reports over the course of a few days.

### Complete Data

Data is considered complete when Akamai has received at least 99% of edge server log data for the selected CP code in the selected date range at the time the report is generated.

Akamai will add the remaining 1% of the data as the server logs are received. Therefore, the same complete report can include slightly more traffic at a later time. This difference will be especially apparent in high-traffic reports. If you need a higher level of precision in your report data, run the report four to five days after the data is marked complete.

### Incomplete Data

Akamai processes log data for URL, and Visitor reports from thousands of servers every day. There is always a lag between when the first and last log file for a particular calendar day is received and processed. Data is considered incomplete when less than 99% of the edge servers logs for an hour period are included in the report, and does not indicate

network problems or a decline in traffic. If you request a report that contains incomplete hours, Akamai will not estimate the data, but will present the incomplete data as is. For example, the traffic on a unique visitors graph will appear to taper off during the incomplete period of time, when in fact it was steady.

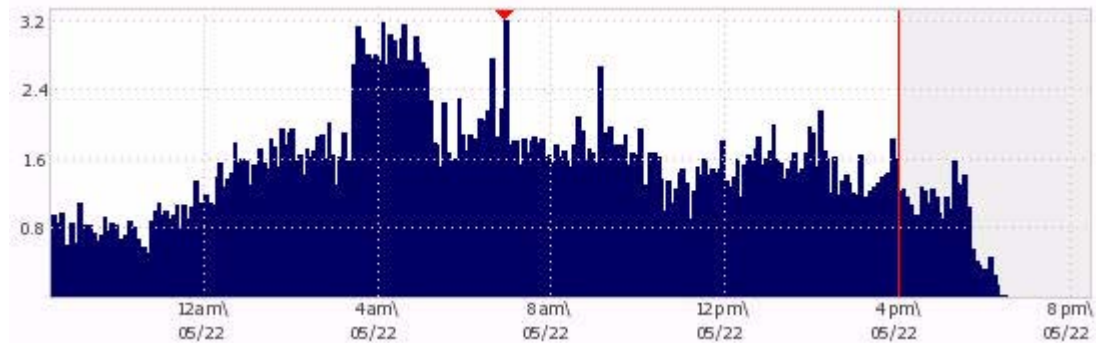


Figure 5.1: Incomplete data

Akamai will always show incomplete data in reports when it is available. Akamai shows estimated data if it is available instead of incomplete data.

## Real-Time Feeds

The two data collection systems serve different purposes. The real-time system is optimized for fast response time, at the expense of data completeness. In other words, the real-time system can only afford to retry contact with a particular edge server a limited number of times within its real-time constraints. If transient Internet connectivity problems prevent contact, the real-time system will do without data from that particular edge server.

Once the real-time system records a value, it is never updated. The recorded value can be considered a snapshot in time until it is “replaced” by a value from the historical, log-based feeds.

## Estimated HTTP Data

Estimated data values are statistical samples taken from a different monitoring system that may over- or under-report data, while historical data comes from completely processed logs. Overall, real-time data values are approximately 90% accurate compared to historical log-based data. Real-time traffic data can be under- or over-reported, due to the complexity of gathering data from multiple data collection servers in real time.

Additionally, real-time data doesn't include any protocol or network overhead; historical log-based data does include these overheads as well as traffic from Akamai's Tiered Distribution cache hierarchy.

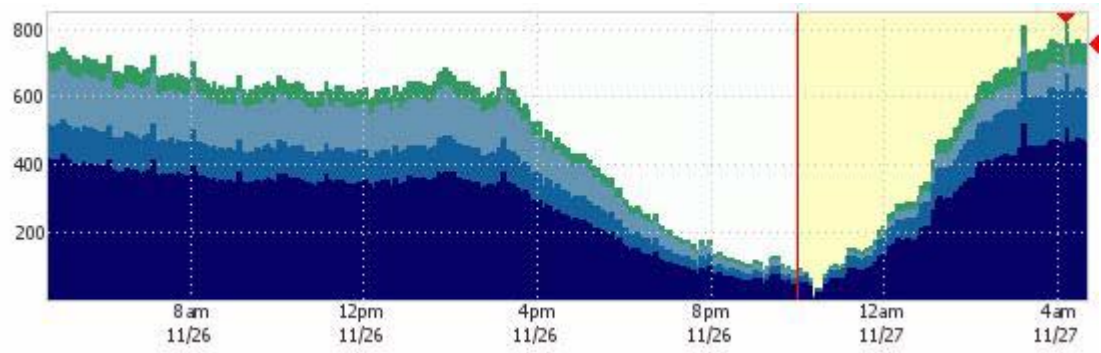


Figure 5.2: Estimated Data

The statistical sampling technique by its nature can show especially large differences for low traffic CP codes. This does not indicate any problem with content delivery; it is due to a small sample taken from a small population.

### Estimated Streaming Data

Note that Live Streams reporting might show a significant discrepancy, where estimated traffic to the right of the estimation line is much higher than the “complete” data to the left of the line. This happens because the streaming server does not send information about a stream until it is complete. Therefore, while 99% of logs may have been received, those logs do not contain information about all the streams currently underway. The discrepancy is especially significant when the stream duration is longer than the log delivery and processing window (one to four hours).

## Traffic Types

### Edge, Midgress, and Origin Traffic

**Edge Traffic** The response traffic served from Akamai edge servers to end users. Also called egress, or Edge egress, traffic.

**Midgress Traffic** The response traffic from Akamai edge servers or the Akamai Tiered Distribution infrastructure to other Akamai edge servers. Midgress bandwidth is counted when the response is sent by an edge server in the Tiered Distribution cache hierarchy and that response is received by another edge server. Midgress traffic can include SureRoute performance traffic incurred in dynamic analysis of best routes for content. Midgress traffic is sometimes referred to as tiered egress traffic.

**Origin Traffic** The traffic from your origin to Akamai edge servers, or from your origin to the Akamai Tiered Distribution cache hierarchy. Also called ingress traffic.

## Request and Response Traffic

### Request Traffic

Request traffic refers to the traffic that is received by Akamai servers as part of a request for content or applications. It includes all request traffic from end user to an Akamai edge server; from an edge server to another edge server, or from an edge server to the Akamai Tiered Distribution Infrastructure; and from Akamai edge servers and/or the Tiered Distribution Infrastructure to your origin. Request traffic includes the Request HTTP header size and any protocol overhead.

### Response Traffic

Response traffic refers to the traffic that is delivered from server to browser. It includes ingress (or origin) traffic, midgress traffic, and egress (or Edge) traffic.



## Traffic Data

The Akamai reporting tools use the following metrics for traffic:

- Hits
- Bandwidth Mbps: Megabits per second rate of bits, averaged over a 5-minute interval
- Volume MB: total sum of Megabytes
- Requests
- Page Views

### Hits

There is no commonly accepted definition of the concept of a “hit” to a Web object or streaming media content. Akamai has settled on the following definitions of a hit.


### HTTP Content Delivery Hit

Reports defines a successful hit as a request to a server that results in a response with an HTTP status code in the 200 or 300 ranges, as well as the Akamai-defined “000” status code. At present, real-time data includes all HTTP status codes in its measure of hits. See “Response Codes” on page 57 for a full description of all HTTP status codes.

This definition of a hit includes requests that were aborted by the end user after Akamai started serving the content. The “000” status code, in particular, arises when an Akamai server was unable to serve even the complete response header, which includes the HTTP status code, before the end user aborted the connection. Requests that result in responses outside these HTTP status code ranges, such as responses in the 400 or 500 ranges, are considered errors, not hits. At present, hits are always an “Edge Egress” metric; that is hits are measured only as transactions between edge-servers and end users.

### HTTP Downloads Hit

For HTTP Download reports, there are two additional metrics defined: Initiated Downloads and Completed Downloads. These metrics are subcategories of hits. An Initiated Download is any hit that contains the first byte of an object and for which the HTTP status code is 200 or 206. A Completed Download is any hit that contains the last byte of an object.

 **Note:** PDF downloads often contain metadata info in the final block (and last byte) of the file, which can be retrieved multiple times in a single request (e.g., in the table of contents). In such an instance, it is possible to see four to five times the completed downloads than initiated downloads.

### Streaming Hit

For Streaming services, successful hits include any response code 0xx, 2xx, 3xx, or 408.

The definition of a successful hit for streaming includes 408 response codes, which can be a client disconnect. This disconnection can be intentional or otherwise on the part of the end user or player. 408 responses can be particularly high for Windows Media Player (all versions) for some of the following reasons:

- Windows Media Services version 4.1 does not send log information for a distribution scenario
- Time out occurs because of bad network conditions
- Client times out because the Windows Media server is overloaded
- Player application is terminated abnormally (e.g., via OS crash/reboot, or the user manually terminating the player with the Task Manager.)
- User loses network connection

- Network problems
- Encoder-related problems. During a live broadcast, if the connection between the Windows Media server and the encoder is lost or the encoder stops streaming for some reason, the server may record 408 logs because clients are disconnected from the server.
- Network Load Balancing using HTTP/1.0 with no affinity specified.
- Fast (Auto) Reconnect failure (code 420) causes 408
- Any of the following players may cause the server to record code 408 logs:
  - Macintosh clients.
  - UNIX clients
  - Certain proxy server and firewall scenarios

## Volume MB - Object Bytes and Overhead Bytes

The process of serving an object via HTTP or streaming protocols includes a number of overheads beyond the raw object size (“object bytes”). Some traffic metrics for MB or Mbps can include these overheads (“overhead bytes”). The chapters on traffic, URL, and visitor data in this document describe whether or not overhead is included in each particular report element.

### HTTP Bytes

The object bytes value for HTTP content is the size of an object in bytes sent from an Akamai edge server to the end user's browser, with status codes 2xx, 3xx, and 000. The bytes of overhead sent via HTTP responses from Akamai edge servers to the end user include HTTP overhead, TCP/IP overhead, SSL overhead, and Ethernet overhead.

Following, we provide brief explanations of each of these overheads. Please consult Appendix A *Detailed Overhead Calculations* for a detailed description of these overheads and how they are computed.

- **HTTP protocol overhead:** bytes sent or received for HTTP request and response headers.
- **SSL protocol overhead:** bytes sent or received as part of the process of exchanging and signing keys.
- **TCP/IP protocol overhead:** bytes sent or received due to TCP/IP packet headers, acknowledgment packets, new connection requests, and packet retransmits.
- **Ethernet protocol overhead:** bytes sent or received for the 14-byte Ethernet packet frames.

### Streaming Bytes

The bytes value for Streaming traffic is the size of the object served between a Streaming edge server and end user's audio/video player. Successful bytes includes response codes 0xx, 2xx, 3xx, and 408). Streaming overhead for streaming traffic between Akamai edge server and end user are the bytes of the requested ARL, TCP/IP overhead, TCP/IP data-packet overhead, TCP/IP acknowledgments overhead, TCP/IP new-connection overhead, UDP/IP overhead, and Ethernet overhead.

*Detailed Overhead Calculations* contains a detailed explanation of these overheads. For the purposes of this section, they are:

- **Streaming protocol overhead:** bytes sent or received for protocol requests and response headers.

- **TCP/IP or UDP/IP protocol overhead:** bytes sent or received for TCP or UDP packet headers, TCP acknowledgments, TCP new connection requests, and packet retransmits.
- **Ethernet protocol overhead:** bytes sent or received for the 14-byte Ethernet packet frames.

## Flash Overhead Bytes

Note that a Flash session may include one or more streams and that Flash is interactive. The session overhead is the difference between bytes served for the session and bytes served for streams within the session. Those overhead bytes include any streaming request bytes, TCP/IP overhead bytes, UDP/IP overhead bytes, and Ethernet bytes. Because a Flash session may include one or more streams and Flash is interactive, a large portion of a streaming bytes number that includes overhead can be for session overhead.

## Compressed File Delivery

Whether an edge server is required to perform any compression or decompression, for example as part of Akamai Last Mile Acceleration (LMA) features, affects how compressed files are measured in Traffic Reports. For example,

- If an edge server retrieves a compressed file of 100 bytes from the origin or NetStorage and delivers a compressed file of 100 bytes to the end user, Traffic reports and Contract Usage reports show 100 bytes.
- If an edge server retrieves uncompressed content of 1000 bytes but is responsible for compressing the content to 100 bytes and delivering it in this state, Traffic reports and Contract Usage reports show 1000 bytes.
- If an edge server retrieves a compressed file of 100 bytes but knows to deliver the uncompressed content of 1000 bytes to the end user, Traffic reports and Contract Usage reports show 1000 bytes.
- If an edge server retrieves a compressed file of 100 bytes from the origin and delivered a compressed file of 100 bytes to the end user, but is responsible for uncompressing the file to 1000 bytes, for example, to process ESI, Contract Usage shows the 1000 bytes to account for the extra CPU requirements for expanding and recompressing the file.

## Concurrent Streams

Live streams and on-demand streams shows the concurrent streams graphed By Average and By Max:

- **By Average:** This graph shows the average number of concurrent streams for each five minute period. The average number of concurrent streams over the time period selected is noted, as well as the latest recorded number of concurrent streams.
- **By Max:** This graph shows the maximum number of concurrent streams for each five minute period. The peak number of concurrent streams over the time period selected is noted, as well as the latest recorded number of concurrent streams.

Note that the maximum number of streams for each five-minute period will often be higher than the average, so the maximum value for the entire period may be higher than the peak point on the graph:

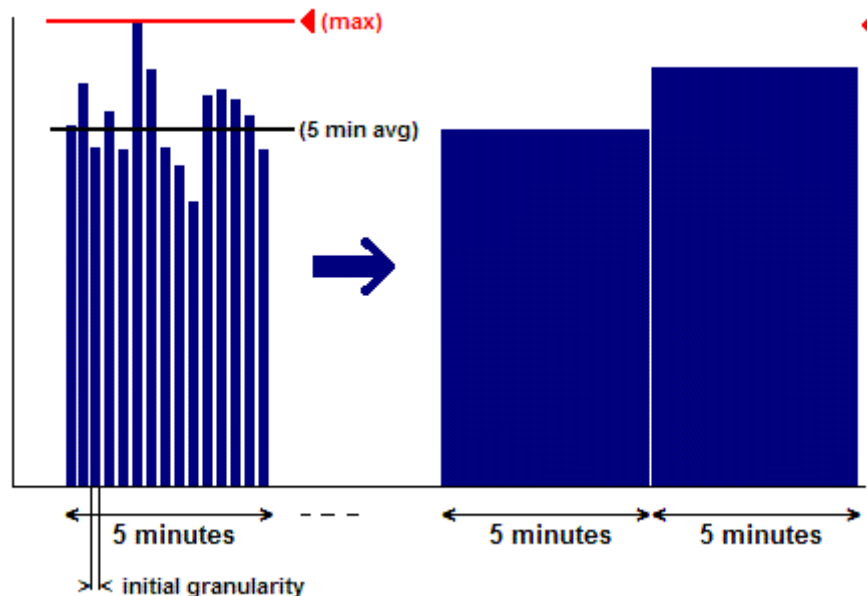


Figure 5.3: Five-minute max values may not be indicative of actual max value

Concurrent streams totals are calculated differently depending on whether one CP code or all CP codes, or several CP codes, is selected for the report:

- For a **single CP code** and **All CP codes**, the maximum and average concurrent streams are exact and based on the start and end time for each individual stream. One-second granularity data is available, and used, when building the single and all CP codes data files.
- When combining **multiple CP codes** the samples file needs is used. The samples file is the average number of concurrent streams occurring during each ten-second interval. To determine the maximum number of concurrent streams when combining multiple CP codes, the average concurrent streams for the selected CP codes is aggregated on a ten-second basis. The maximum is then taken from ten-second time period totals.

## Flash Concurrent Connections

Real-time Flash concurrent connections can sometimes appear much higher than historical Flash concurrent connections. The Flash Media Server will hold any idle connection open for a default 60 minutes. An idle connection can be caused by an end user pausing a stream, or by a player being closed without having called an explicit connection close. The idle connections can only be distinguished from active connections in the historical logs, which causes the discrepancy. The shorter the Flash content the more idle connections and the higher the discrepancy. The idle connection timeout can be reduced for your Flash content to minimize this discrepancy. Be aware that this will reduce the time a stream can be paused and resumed without restarting the connection. If you want to do this please contact your Akamai representative.

## Requests

For EdgeComputing services, any request that results in a response from the edge server to the end user is considered a request. In contrast to HTTP, this definition includes requests that result in errors in the 500 range and below.

## Page Views

For services like Web Application Accelerator, Site Delivery, and Site Accelerator, page views includes all pages where contenttype=text/html from the edge server to the end user. 404 Not Found errors are no longer included in the page view tally for traffic reports or contract usage.

# URL Data

URL data may not be included for all services; additional fees may apply. Contact your Akamai Account Manager for more information.

In the EdgeControl Management Center UI, you can see up to 500 URLs. Downloading the report in CSV format will provide the entire list of URLs.

## 50-Hit Requirement

HTTP URL data is stored only for URLs with at least 50 hits per day. This means that URLs with less than 50 hits per day will be excluded from single day reports and reports that span multiple days will under-report the traffic for some URLs.

The 50 hit requirement is in place to prevent customers with a large number of distinct URLs with few hits from overwhelming the reporting systems. It is an important safety mechanism to ensure the availability of reports for all users.

The example below demonstrates how a multi-day report for a single URL can under-report the number of hits for a low-traffic URL:

Table 5.1: Multi-Day URL Data Under-Reported for Low-Traffic URL

	OK Hits	Error Hits	Total Hits
Monday:	300	10	310
Tuesday:	180	20	200
Wednesday:	35	5	40
Thursday:	40	20	60
Friday:	20	5	25

Table 5.1: Multi-Day URL Data Under-Reported for Low-Traffic URL

	OK Hits	Error Hits	Total Hits
<b>Actual Total:</b>	575	60	635
<b>URL Report Total:</b>	520	50	570

In this example, the traffic on Wednesday and Friday is not included in the total number of hits in the URL report. The volume, origin hits, and all other data columns for these URLs will be similarly under-reported.

The 50-hit threshold combines both OK, or successful, and error hits. Therefore, you may see a URL with less than 50 OK hits in your report. URLs with less than 50 OK hits can have sufficient error hits to exceed the 50-hit limit.

The 50-hit requirement does not apply to Traffic or Visitor reports. The traffic for all URLs, regardless of number of hits, is included in these reports.

## HTTP URL Hit Limit

HTTP URL data collection is limited to 50 million URLs a day. The limit has been set to optimize data collection and achieve efficiencies. If data volume exceeds the limit, collection is automatically suspended. In the unlikely event of data volume nearing the limit, Akamai will proactively help you appropriately configure your service to avoid automatic suspension.

## Filters

Using filters in the current reports, you can reduce the URL paths displayed in the URL reports to just the files or just the directories and/or reduced by string or regular expression matching. You can now search the URL reports by filtering on the following options:

- A URL that contains, starts with, or ends with a specific substring
- A URL that doesn't contain, doesn't start with, or doesn't end with a specific substring
- A URL that matches or doesn't match a regular expression
- A URL or search for an exact URL.

You can choose to have the search be case sensitive or case insensitive.

## ESI Fragments

Customers using Edge Side Includes (ESI) will see fragments appearing in URL reports. By design, the traffic attributed to fragments in the URL reports is not included in any other hits and bits data because it does not represent actual data delivered to end users. Fragments are incorporated into container pages as part of the dynamic content assembly process, and it is these container pages that are actually served to end users. By including fragments in URL reports, however, ESI content providers get an easy way to gauge the frequency of fragment use, but they lose the ability to draw comparisons between URL reports with other reports.

## Aggregation

You can choose to aggregate the data in URL reports by clicking More Options. For the time period selected, you will see daily totals for the URL metric you select. You can also choose to see total daily aggregations.

## Flash URLs

The Adobe FLVPlayer component tries to connect in different ways, with and without the file extension, to ensure that the connection is successful. Akamai reports does not recognize optional file extensions and the strings making up the URLs are different, so the URL will show up twice in a report, once with the file extension and once without the extension.

You can choose to

- ignore all the 404 errors for that file. That number will be bloated because the player will try to connect in a way that might not work.
- change your player to a more streamlined one that does connection in only one way.

## URL Data in HTML format

In EdgeControl interface, long URLs may be truncated using an ellipsis in the middle of the URL. The entire URL can be seen by downloading the report in CSV format.

## URL Data in CSV format

When you choose to download URL data in CSV format, the file will include data for all URLs, not just the top 500.

# Visitors Data

Visitor data may not be included for all services; additional fees may apply. Contact your Akamai representative for more information.

## Unique Visitors

By default, unique visitors in Content Delivery are defined as the number of unique combinations of client IP addresses plus User Agent string. There is no concept of a timeout.

Unique visitors in On Demand Streaming are defined as having a unique player ID or when not present, a unique client IP address. Player IDs (sometimes called a GUID) are not present when not reported by the client or a well-known anonymous value (e.g., anything starting with 3300AD50-2C39-46c0-AE0A- for Windows Media) is reported. To avoid overreporting, all anonymous PlayerIDs for a single report period are treated as one PlayerID.

Note that the number of visitors displayed does not accurately represent the number of actual people visiting your site. Many people can appear as a single IP address by sharing proxies, caches, NAT firewalls or even simply sharing the same computer at home. Users who work for a company with a locked-down computer image will appear to have the same user agent strings. Other reasons for undercounting of unique visitors are clients operating through anonymizers.

One person can also appear as many IP addresses by using dynamic IP addressing (most dialup and PPPoE users), being load balanced across proxies and caches or simply using multiple computers (e.g., at home and at work). A user who has lots of spyware and plugins may be more likely to have multiple unique user agent strings. Other reasons for overcounting of unique visitors may include: robots; rogue client software that keeps changing its ID string; users that upgrade software or use multiple client software agents.



### Visitor Data Prerequisite

User agent strings must be logged in order for the visitors page to work correctly. This feature (Edge-control: log-user-agent in metadata) can be turned on using the Reporting Options in Configurations on the EdgeControl Management Center, if you have access, or your Akamai representative can turn it on for you. User agent data is only collected from the time this feature is turned on.

Player information in the Streaming Visitors reports is automatically included in streaming log files.

### Visitor Data Granularity

A Visitor report shows data for a period of a day. Besides the significant costs in tracking unique visitors for any longer period of time, there are reasons why it is not useful to track them longer with IP Address and User Agent string. For example, there are dialup users that get a different IP every time they connect. Note that IP addresses are tracked per hour, so two hits from the same IP at 1:59:59 and at 2:00:01 would result in separate IP addresses in separate hours, counted as two unique visitors.

In addition, unique visitor information can't be aggregated beyond the intervals supplied by Akamai. For example, if Akamai provided five-minute reports for unique visitors, a single visitor who visited once a minute would be reported once in each hypothetical five-minute interval. The same user would be reported just once in a daily report. If the hypothetical five-minute reports had been aggregated, they would have claimed 288 unique visitors instead of the actual single unique visitor.

### 'Other USA' Data in US States List

The Other USA category in a Visitors report User Location - US States list includes primarily AOL requests in the United States, since AOL provides only a country with a user's IP address, not state-level information.

## Data Granularity

Currently, most data in portal reports is five-minute data, which is what is displayed in the HTML reports as well as delivered in CSV reports.

FTP data is hourly. URL and unique visitor data is daily.

## Timezones

URL and visitor reports use the timezone associated with the CP code for which the report was run. Traffic reports use the timezone specified in the user profile of the user running the report.

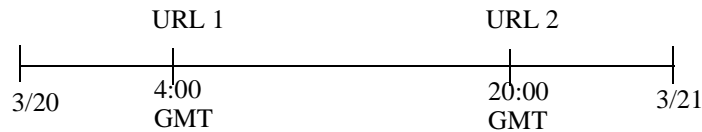
### Important Considerations for Configurable Timezones

Timezones can be configured on a cp code and user basis. Since traffic reports are based on the portal user's configured timezone, and URL and visitor reports are based on the time zone, data for the same traffic can appear to occur at different times when comparing reports. Consider the following example:

CP code 1 - timezone GMT-12  
CP code 2 - timezone GMT+6  
User 1 - timezone GMT +12

Note that for these examples the timeline for hits on URLs 1 and 2 is displayed from the perspective of the CP codes and user each having different timezones.

Say the URL hits occur as follows:

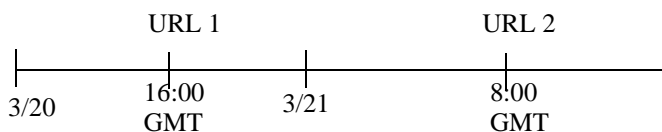


URL Report (using cp code timezone basis):

- Viewing URL report for CP code 1 (GMT-12), URL 1 appears on 3/19 and URL 2 appears on 3/20. (URL 1 at 4:00 GMT on 3/20 shifts -12 hours to 3/19 at 16:00 GMT, and URL 2 at 20:00 GMT on 3/20 shifts -12 hours to 8:00 GMT on 3/20.)
- Viewing the URL report for CP code 2 (GMT +6) URL 1 appears on 3/20 and URL 2 appears on 3/21 (URL 1 at 4:00 GMT on 3/20 shifts +6 hours to 10:00 GMT on 3/20, and URL 2 at 20:00 GMT on 3/20 shifts +6 hours to 2:00 GMT on 3/21. )
- Viewing the combined URL report for both CP codes for the 24-hour period of 3/20 shows URL 2 for CP code 1, and URL 1 for CP code 2 , since URL 1 shifts -12 hours to 3/19 for CP code 1, and URL shifts +6 hours to 3/21 for CP code 2.

The traffic report, using the user timezone basis, would show traffic for those same hits as follows:

Traffic report for user (timezone GMT +12)




The hits for URL 1 and 2 can appear to occur on different days depending on whether a report is run for one CP code or another, and depending on which report is viewed. Running one URL report for multiple CP codes with different timezone offsets over a short time range could exclude data for the same URL.

To avoid these apparent discrepancies, either keep GMT as the setting for all CP codes and users, or ensure that the configured offset is consistent between CP codes and users.

## Data Latency<sup>1</sup>

Data latency depends on a number of factors like log collection and log processing time.

 **Note:** Data latency can change with network or processing changes. This information is provided for informational purposes only.

With Akamai, there are logs on potentially thousands of machines. Akamai attempts to retrieve the correct log line for each customer from each machine from which the content may be served.

The data from those potentially thousands of machines can take time to compile. Hits and bytes data combine easily, and this information is available quickly in your reports, most likely within a four-hour window. However, URL data can take significantly longer time to be available because of the volume of data; there is the potential for a unique URL to come from each log line. Summary URL data can be available most likely within five to ten hours. Detailed URL data can average a 20-hour delay, and visitor data can take longer than 20 hours.

## Customized Reports

### Date Ranges

For traffic and URL reports you can select a time period of one day (default), two days, seven days, or you can customize a date range. Data is available for 90 days in the past. If you are interested in retaining data for more than 90 days, Akamai suggests you set up recurring reports and store the data that is emailed to you.

Visitor reports data can be seen for a period of one day.

### Multiple CP Codes

You can select a CP code for which to run a report, or for Visitor reports you can choose “All CP Codes” or a single CP code.

To customize a report for more than one CP code, clicking More Options in the UI will give you a field where you can CTRL-click multiple CP codes, combining their data into a single report.

## Recurring Email Reports

You can email a report by clicking More Options at the top of a report page. You can define the following for an emailed report:

- Report Name - A name you provide for the report. Indicating the report type and recurrence in the name might be helpful.
- Email addresses: The email addresses to receive the report, up to ten; multiple email address must be separated by comma, semi-colon, or line break.
- Send: The format of the report to be sent, either HTML or CSV.

---

1.

- **Recurrence:** The frequency at which the report will be sent, including Send Now (instant), Daily, Weekly (Sunday to Saturday), Weekly (Monday to Sunday), or Monthly.

The e-mailed report will have a subject line of "Emailed Report." In the e-mail, the header identifies the CP code description and number, the Akamai service, as well as the time range the report data covers. The person who configured the e-mailed report is also identified by first and last name, and e-mail address if it's available:

Includes travel.example.com (3459), finance.example.com (3483)

Service: HTTP Content Delivery

Covers from Jan. 1, 2006 12:00AM to Jan. 6, 2006 11:59PM

Sent by Alison Smith, asmith@example.com

Once you've set up a recurring report, you can review the configuration by clicking **Recurring Reports** in the left-hand navigation menu. The **Recurring Reports** table displays the name, recurrence frequency, and the e-mail address(es) of the recipient(s) of all recurring reports.

Recurring reports are sent only when the data is 99% complete.

## Recurring report details

The **Report Details** page provides information about a particular recurring report. From here you can edit the details of the report and its data, or its recurrence options.

### Report Details

- **Service:** The Akamai service for the report, for example, HTTP Downloads.
- **Report Type:** The report included in the e-mail, for example, Traffic Over Time.
- **Content:** The CP codes and names included within the report.
- **Protocol:** Whether the report contains SSL traffic, non-SSL traffic or both.
- **Tag:** The name of the tag used to filter the report, or None.
- **Report Format:** The format of the report e-mail, either HTML or CSV.

### Recurrence Options

- **Report Name:** The unique name you gave the report when it was created.
- **E-mail Address:** The e-mail address of the person who created the e-mail.
- **Recurrence:** The frequency with which the report is sent, either Daily, Weekly, or Monthly.

## Supported Email Clients

Supported email clients for HTML reports includes:

- Microsoft Outlook 2000
- Microsoft Outlook 2002
- Microsoft Outlook 2003
- Thunderbird

Outlook 2007, Outlook Web Access, Lotus Notes, or most of the web-based clients such as Yahoo or Gmail, are not supported as they do not support advanced HTML.

## Suspended Report Delivery

You can suspend or resume delivery for a recurring report by clicking the corresponding Resume Delivery or Suspend link for the report in the Recurring Reports table. Note that only active reports count toward your recurring reports quota. Suspended reports are not counting toward your quota.

Recurring reports are sometimes suspended automatically, for reasons such as the following:

- EdgeControl Management Center access changes to Reporting Groups
- Changes to contract
- Invalid parameters entered within the scheduled report by user
- Changes to the service(s) associated with a CP code
- EdgeControl Management Center access changes for a CP code and user

If necessary, you can modify your scheduled report to address such issues, then save the changes to reinstate delivery.



# Glossary

<b>cache hit</b>	The number of requests served by Akamai Edge Servers without having to go to the origin. Cache hits are calculated as (edge_hits minus origin_hits) and are not derived directly from logs.
<b>CIR</b>	Committed Information Rate, the maximum rate of traffic that an Akamai customer can serve during a month without incurring additional charges; measured in units of Mbps.
<b>CIV</b>	Committed Information Volume, the maximum amount of traffic that an Akamai customer can serve during a month without incurring additional charges; measured in units of MB or GB.
<b>completed download</b>	An HTTP transaction in which the last byte of the object was served to the end user. This will include transactions in which the entire object was served all at once (status code 200) as well as transactions in which the last byte was served in response to a byte-range request (status code 206).
<b>connection</b>	An underlying TCP-level socket connection set up between a client and a server. The Competitive Benchmark report uses this metric.
<b>content type</b>	A header that describes the file type that follows, which the browser uses to render the content properly. See also MIME.
<b>CP</b>	Content Provider, an Akamai customer serving traffic on the Akamai network.
<b>CP code</b>	A numeric identifier unique to a content provider and used to distinguish their traffic for reporting and invoicing purposes.
<b>download traffic</b>	The GET/HEAD traffic on a Fast File Upload (for WAA or DSA) CP code. See Also upload traffic.
<b>DST</b>	Daylight Savings Time. The EdgeControl Management Center does not recognize Daylight Savings Time.
<b>edge server</b>	One of Akamai's up to 20,000 machines spread through more than 1,000 networks in more than 70 countries; edge servers deliver Web and Streaming traffic to end users.
<b>edge traffic</b>	The response traffic served from Akamai edge servers to end users. Also called egress, or edge egress, traffic.
<b>egress traffic</b>	The response traffic served from Akamai edge servers to end users. Also called Edge traffic.
<b>end user</b>	The final recipient of content delivered by the Akamai network; generally a Web browser or a media player.
<b>ESI</b>	Edge Side Includes, a mechanism for generating dynamic content on edge servers by assembling pages out of fragments customized to each end user.
<b>GB</b>	Gigabyte, 1 billion bytes $1 \text{ billion} = 1,000,000,000 = 10^9$
<b>Gbps</b>	Gigabits per second, a unit of transfer rate equal to 1 billion bits per second.

<b>geographic area</b>	The location (state or country or province) of the user requesting content. This data is generated by using the client IP address to look up the location in an EdgeScape database on the Akamai edge server.
<b>GMT</b>	Greenwich Mean Time, the time in Greenwich, England, which is the universal standard for time. At midnight, GMT, Eastern Standard Time is 7 p.m. and Pacific Standard Time is 4 p.m.
<b>HTTP</b>	Hypertext Transfer Protocol, the application protocol used to deliver Web objects from servers to browsers.
<b>ingress traffic</b>	See origin traffic.
<b>initiated download</b>	An HTTP transaction that must include the first byte of the file, and is either HTTP status code 200 (successful) or 206 (successful byte range request).
<b>Java RMI</b>	Java Remote Method Invocation; similar to a remote procedure call but with the ability to pass objects, RMI enables objects on different computers to interact in a distributed network.
<b>JDBC</b>	Java Database Compatibility, an API for enabling programs written in Java to connect with popular databases to retrieve and post data.
<b>LDS</b>	Log Delivery Service, Akamai's service for delivery of detailed, per-hit server logs to content providers.
<b>LMA</b>	Last Mile Acceleration, Akamai's implementation of dynamic protocol optimization, including compression, for bandwidth-limited end users.
<b>MB</b>	Megabyte, 1 million bytes 1 million = 1,000,000 = $10^6$
<b>Mbits/sec</b>	Megabits per second, a unit of data transfer rate equal to 1,000,000 bits per second.
<b>Mbps</b>	See Mbits/sec.
<b>midgress traffic</b>	The response traffic from Akamai edge servers or the Akamai Tiered Distribution infrastructure to other Akamai edge servers. Midgress bandwidth is counted when the response is sent by an edge server and when the response is received. Midgress traffic can include SureRoute performance traffic incurred in dynamic analysis of best routes for content, as well as Site Shield Traffic. It is sometimes referred to as tiered egress traffic.
<b>MIME</b>	Multipurpose Internet Mail Extensions, an official Internet standard that enables the exchange of different file types over the Internet. MIME identifies the content and format of a file exchanged on the Internet. When the browser recognizes the content type, it opens the file with an appropriate application that's built into the browser, or the browser launches the appropriate application or plug-in to open the file. An example of a content type is text/html, which means the browser opens the text file in HTML.
<b>origin</b>	Or Origin Server. A content provider's server where Web objects and streaming media originates before being cached in the Akamai network.
<b>origin hit</b>	A request for which Akamai edge server cache did not have the requested content or the requested content was not fresh, and therefore needed to retrieve the content from the origin.
<b>origin traffic</b>	The traffic from your origin to Akamai edge servers, or from your origin to the Akamai Tiered Distribution cache hierarchy.



<b>page</b>	A Web page with content MIME type of "text/html" identified by a URL. A page excludes objects on that page, such as an image.
<b>page view</b>	The delivery of a file by Akamai that has a "text/html" content type but excludes redirects (HTTP response code 301/302) and File Not Found error page (HTTP response code 404).
<b>PB</b>	Petabyte, 1 quadrillion bytes 1 quadrillion = 1,000 terabytes = 1,000,000,000,000,000 = $10^{15}$
<b>request header</b>	A part of an HTTP request from a browser to a server; the request header is used to describe the browser's capabilities, such as the languages and encodings accepted by the browser.
<b>request traffic</b>	The traffic that is received by Akamai servers as part of a request for content or applications. It includes all request traffic from end user to an Akamai edge server; from an edge server to another edge server, or from an edge server to the Akamai Tiered Distribution Infrastructure; and from Akamai edge servers and/or the Tiered Distribution Infrastructure to your origin. Request traffic includes the Request HTTP header size and any protocol overhead.
<b>response header</b>	A part of an HTTP response from a server to a browser; the response header includes information such as caching and authentication directions.
<b>response traffic</b>	The traffic that is delivered from server to browser. It includes ingress (or origin) traffic, midgress traffic, and egress (or Edge) traffic.
<b>site activity</b>	The number of new HTTP connections to a site in a five-minute period, used in the Competitive Benchmark report.
<b>SOAP</b>	Simple Object Access Protocol, an HTTP- and XML-based protocol that enables communication between disparate applications.
<b>SSL</b>	Secure Socket Layer, an application level protocol for encrypting Web data for secure, end-to-end communication between servers and browsers.
<b>SureRoute</b>	Akamai's system for dynamic best route analysis for the delivery of Web content.
<b>TB</b>	Terabyte, 1 trillion bytes 1 trillion = 1,000,000,000,000 = $10^{12}$
<b>TCP</b>	Transmission Control Protocol, an Internet protocol that provides reliable byte-stream connections; application protocols like HTTP are built on top of TCP.
<b>throughput</b>	The volume of data moved successfully from one point to another in a given time period.
<b>Tiered Distribution</b>	A hierarchical content distribution network used to improve performance and decrease load on the origin server.
<b>tiered egress traffic</b>	See midgress traffic.
<b>UDP</b>	User Datagram Protocol, an Internet protocol that provides delivery of data packets; UDP is often used in Streaming protocols.
<b>upload traffic</b>	The POST/PUT traffic on a Fast File Upload (for WAA or DSA) CP code. See Also download traffic.
<b>Web site</b>	One domain and up to 10 hostnames, as defined by Akamai. For example, in www.example.com and news.example.com, "example" is the domain and "www" and "news" are hostnames.



## **XSLT**

Extensible Stylesheet Language Transformations, a standard-based language used to transform the structure of an XML document into another format (e.g., XML, HTML, WML).

# Appendix A. Detailed Overhead Calculations

## In This Appendix:



- ◆ Packets • 43
- ◆ HTTP • 44
- ◆ Streaming • 48

The total bytes required to serve a Web object or streaming media file include the size of the object as well as a number of protocol overhead components. This section describes the calculation of the overheads.

## Packets

First, it is critical to understand the data served over the Akamai network is broken down into TCP or UDP packets that are subsequently transmitted over Ethernet. UDP packets, often used in streaming protocols, have two forms, the “first” packet in transaction and the “subsequent” packets. One of the important overheads discussed below are the headers for these packets that are critical for routing the packet through the Internet.

# HTTP

This document section is provided for informational purposes only and shall not be construed as providing any representation or guarantee as to the matters discussed. Akamai assumes no obligation to update or correct any matters discussed in the document.

This section provides the constants and equations used to calculate HTTP total and overhead bytes, and detailed descriptions for these constants and equations.

## Constants and Equations

The following alphabetical list of constants and equations are used for HTTP overhead byte calculations. Descriptions of these components follow the list.

ack packet for HTTP client request	= TCPIPOVERHEAD + ETHERNETOVERHEAD
bytes of request	= number of characters in the URL
DEFAULTHTTPOVERHEAD	= 304 bytes
ETHERNETOVERHEAD	= 14 bytes
incoming ack packets	= (TCPIPOVERHEAD + ETHERNETOVERHEAD) * packets / 2
incoming syn, ack, finack, and reset packets	= 4 * (TCPIPOVERHEAD + ETHERNETOVERHEAD)
outgoing synack, and fin packets	= 2 * (TCPIPOVERHEAD + ETHERNETOVERHEAD)
outgoing TCP overhead packets	= (TCPIPOVERHEAD + ETHERNETOVERHEAD) * packets
PACKETSIZE	= 1500 bytes
size of object	= object bytes
TCPIPOVERHEAD	= 40 bytes

### Ack packet for HTTP client request

An acknowledgment is an empty TCP/IP packet, i.e., a packet with a header but no data, used to signal the receipt of data or the receipt of a request for data. The overhead associated with sending or receiving each acknowledgment is 40-byte TCP/IP packet header. A server sends an acknowledgment when it receives a request for data (i.e., the object and its associated response header), and it receives an acknowledgment when it requests data. This request acknowledgment incurs an overhead of 40 bytes.

### DEFAULTHTTP OVERHEAD

The default value is 304 bytes, but there is some variability according to what is being requested and how individual browsers make the requests.

Streams that use the HTTP protocol will have HTTP overhead. This value consists of two components, both extracted directly from the Akamai server logs:

- **Bytes of the HTTP request:** request overhead is only used in Contract Usage reports for certain services.
- **Bytes of the HTTP response headers:** the response headers are part of the HTTP protocol and contain information about the transaction, such as downstream caching or TTL directives.

## ETHERNET OVERHEAD

Each TCP/IP packet is wrapped in an Ethernet frame with an Ethernet frame header of 14 bytes (ETHERNETOVERHEAD). To determine the overhead for all frames, multiply the total number of TCP/IP packets by 14. The result is that every packet sent or received incurs a total of 54 bytes of overhead: 40 for the TCP/IP header and 14 for the Ethernet header.

It is important to emphasize that whether the packet is sent or received by an Akamai server determines how it is accounted in the traffic statistics. Packets sent by Akamai servers are part of Edge or Tiered Egress traffic. Packets received by Akamai servers are part of Ingress or Request traffic.

## Incoming ack packets

An acknowledgment is an empty TCP/IP packet, i.e., a packet with a header but no data, used to signal the receipt of data or the receipt of a request for data. The overhead associated with sending or receiving each acknowledgment is 40-byte TCP/IP packet header. A server sends acknowledgments when it receives data, and it receives acknowledgments when it sends data. In data transfers involving more than a handful of packets, one acknowledgment is usually generated for every two TCP/IP packets of data.

For example, to calculate the number of acknowledgments received by an edge server sending an object (and its associated response header) to an end user, you would first round the number of TCP/IP packets for sending the data (calculated above) to the nearest highest even number. Then, you would divide this number by 2 to get the number of acknowledgments received. Multiplying by 40 yields the number of acknowledgment overhead bytes incurred when the end user's browser acknowledges to the edge server that the data was received.

## Incoming syn, ack, finack, and reset packets

The opening and closing of a new connection between the end- user's browser and the edge server involves the sending of six empty TCP/IP packets. The TCP/IP new-connection overhead associated with sending each packet is its 40-byte header. Four of these packets are sent by the browser to the edge server (incoming syn, ack, finack, and reset packets), and the other two are sent by the edge server to the browser (outgoing synack and fin packets). The overhead involved in sending these packets is added only when the browser makes its first (or only) request to the edge server. Subsequent requests on the same connection are not charged this overhead.

## Outgoing synack and fin packets

The opening and closing of a new connection between the end- user's browser and the edge server involves the sending of six empty TCP/IP packets. The TCP/IP new-connection overhead associated with sending each packet is its 40-byte header. Four of these packets are sent by the browser to the edge server (incoming syn, ack, finack, and reset packets), and the other two are sent by the edge server to the browser (outgoing synack and fin packets). The overhead involved in sending these packets is added only when the browser makes its first (or only) request to the edge server. Subsequent requests on the same connection are not charged this overhead.

### **Outgoing TCP overhead packets**

Each data packet consisting of the object being sent (served) or received and its associated HTTP response header is wrapped in TCP/IP packets. The HTTP request is also wrapped in (usually) one TCP/IP packet. Each TCP/IP packet has a TCP/IP header. The number of bytes in the header represents the TCP/IP overhead for the TCP/IP packet. Consequently, to determine the overhead for all packets, first compute the number of TCP/IP packets required to send the data and the request (see “PACKETSIZE”). Since the overhead per TCP/IP packet is the size of its header, a 40-byte structure (“TCPIPOVERHEAD”), multiply the number of packets by 40 to get the total overhead.

### **PACKETSIZE**

Each TCP/IP packet is 1500 bytes. The 1500 bytes consist of 40 bytes for the packet header and 1460 bytes for data. To compute the number of packets required for the data, divide the number of bytes of data by 1460, and round up to the next highest integer. Then, for services that account for Request traffic, add 1 to this integer to account for the TCP/IP packet containing the HTTP request header.

For example, suppose an object and its HTTP response header are 12,000 bytes total (as described above, these numbers are extracted from the log line). Then  $12000/1460$  rounded up to the next highest integer results in a total of 9 TCP/IP packets needed to send this object and its response header. Add 1 to account for the TCP/IP packet containing the HTTP request, which gives a total of 10 packets. Multiply 10 by the number of bytes in the packet header (40) to get the total TCP/IP overhead (400) for sending the data and request.

### **Size of object**

This value is extracted directly from the logs generated by Akamai HTTP servers. This value can be the size of the entire object, or the size of the portion of the object served in a byte-range request. Note that for objects served with the Last Mile Acceleration (LMA) feature, the size of the object used is the size of the object before compression.

### **TCPIPOVERHEAD**

Each data packet consisting of the object being sent (served) or received and its associated HTTP response header is wrapped in TCP/IP packets. The HTTP request is also wrapped in (usually) one TCP/IP packet. Each TCP/IP packet has a TCP/IP header. The number of bytes in the header represents the TCP/IP overhead for the TCP/IP packet. Since the overhead per TCP/IP packet is the size of its header, a 40-byte structure, multiply the number of packets by 40 to get the TCP/IP overhead.

## Overhead Calculations

For log lines with bytes that should be tabulated, protocol can be HTTP or TCP, and the overhead will be calculated in a manner specific to each protocol.

**HTTP** Assuming each TCP packet carries `PACKETSIZE - TCPIPOVERHEAD` data bytes:

estimated number of packets	= (size of object + <code>PACKETSIZE</code> - <code>TCPIPOVERHEAD</code> - 1) / ( <code>PACKETSIZE</code> - <code>TCPIPOVERHEAD</code> )
number of packets	= MAX( <code>s_pkts_sent</code> , <code>c_pkts_received</code> , estimated number of packets )
estimated outgoing bytes	= size of object + ack packet for HTTP client request + outgoing TCP overhead packets + outgoing synack, and fin packets
estimated incoming bytes	= bytes of request + <code>DEFAULTHTTPOVERHEAD</code> + ack for HTTP client request + incoming ack packets + incoming syn, ack, finack, and reset packets

# Streaming

This section provides the constants and equations used to calculate total streaming bytes for Windows Media, Flash, QuickTime, and Native Real Media, detailed descriptions for these constants and equations, and the differences in the overhead calculations for streams using HTTP, UDP, and TCP protocol.

## Constants and Equations

The following alphabetical list of constants and equations are used for streaming overhead byte calculations. Descriptions of these components follow the list.

<b>DEFAULTHTTPOVERHEAD</b>	= 304 bytes
<b>ETHERNETOVERHEAD</b>	= 14 bytes
<b>IPOVERHEAD</b>	= 20 bytes
<b>PACKETSIZE</b>	= 1500 bytes
<b>TCPIPOVERHEAD</b>	= 40 bytes
<b>UDPOVERHEAD</b>	= 8 bytes
<b>ack packet for HTTP client request</b>	= TCPIPOVERHEAD + ETHERNETOVERHEAD
<b>bytes of request</b>	= number of characters in the URL
<b>size of stream</b>	= MAX( sc_bytes, c_bytes )
<b>number of packets (WMS/QT)</b>	= reported number of packets. For more information, see "Number of packets" description below.
<b>estimated number of packets (HTTP/TCP)</b>	= (size of stream + PACKETSIZE - TCPIPOVERHEAD - 1) / (PACKETSIZE - TCPIPOVERHEAD)
<b>estimated number of packets (UDP)</b>	= 1 + (size of stream + UDPOVERHEAD - 1) / (PACKETSIZE - IPOVERHEAD)
<b>reported number of packets</b>	= MAX( s_pkts_sent, c_pkts_received, estimated number of packets )
<b>number of packets (Native Real Media)</b>	= reported number of packets. For more information, see "Number of packets" description below.



estimated number of packets (HTTP/TCP)	$= (\text{size of stream} + \text{PACKETSIZE} - \text{TCPIPOVERHEAD} - 1) / (\text{PACKETSIZE} - \text{TCPIPOVERHEAD})$
estimated number of packets (UDP)	$= 1 + (\text{size of stream} + \text{UDPOVERHEAD} - 1) / (\text{PACKETSIZE} - \text{IPOVERHEAD})$
reported number of packets	$= \text{MAX}(\text{c\_pkts\_received}, \text{estimated number of packets})$
<b>number of packets (Flash)</b>	<i>For more information, see "Number of packets" description below.</i>
reported number of packets	$= (\text{size of stream} + \text{PACKETSIZE} - \text{TCPIPOVERHEAD} - 1) / (\text{PACKETSIZE} - \text{TCPIPOVERHEAD})$
<b>incoming ack packets</b>	$= (\text{TCPIPOVERHEAD} + \text{ETHERNETOVERHEAD}) * \text{number of packets} / 2$
<b>incoming syn, ack, finack, and reset packets</b>	$= 4 * (\text{TCPIPOVERHEAD} + \text{ETHERNETOVERHEAD})$
<b>outgoing synack, and fin packets</b>	$= 2 * (\text{TCPIPOVERHEAD} + \text{ETHERNETOVERHEAD})$
<b>outgoing TCP overhead packets</b>	$= (\text{TCPIPOVERHEAD} + \text{ETHERNETOVERHEAD}) * \text{number of packets}$

## DEFAULT HTTP OVERHEAD

The default value is 304 bytes, but there is some variability according to what is being requested and how individual browsers make the requests.

Streams that use the HTTP protocol will have HTTP overhead. This value consists of two components, both extracted directly from the Akamai server logs:

- **Bytes of the HTTP request:** request overhead is only used in Contract Usage reports for certain services.
- **Bytes of the HTTP response headers:** the response headers are part of the HTTP protocol and contain information about the transaction, such as downstream caching or TTL directives.

## ETHERNET OVERHEAD

Each TCP/IP packet is wrapped in an Ethernet frame with an Ethernet frame header of 14 bytes (ETHERNETOVERHEAD). To determine the overhead for all frames, multiply the total number of TCP/IP packets by 14. The result is that every packet sent or received incurs a total of 54 bytes of overhead: 40 for the TCP/IP header and 14 for the Ethernet header.

It is important to emphasize that whether the packet is sent or received by an Akamai server determines how it is accounted in the traffic statistics. Packets sent by Akamai servers are part of Edge or Tiered Egress traffic. Packets received by Akamai servers are part of Ingress or Request traffic.

#### **IPOVERHEAD**

If the protocol for sending the data is UDP, then the UDP/IP overhead consists of only one component: the UDP/IP data-packet overhead. Each data packet of the object being served is wrapped in UDP/IP packets. The first UDP/IP packet has both an IP header of 20 bytes and a UDP header of 8 bytes. Subsequent packets have only an IP header of 20 bytes.

#### **PACKETSIZE**

Each TCP/IP packet is 1500 bytes. The 1500 bytes consist of 40 bytes for the packet header and 1460 bytes for data. To compute the number of packets required for the data, divide the number of bytes of data by 1460, and round up to the next highest integer. Then, for services that account for Request traffic, add 1 to this integer to account for the TCP/IP packet containing the HTTP request header.

For example, suppose an object and its HTTP response header are 12,000 bytes total (as described above, these numbers are extracted from the log line). Then  $12000/1460$  rounded up to the next highest integer results in a total of 9 TCP/IP packets needed to send this object and its response header. Add 1 to account for the TCP/IP packet containing the HTTP request, which gives a total of 10 packets. Multiply 10 by the number of bytes in the packet header (40) to get the total TCP/IP overhead (400) for sending the data and request.

#### **TCPIPOVERHEAD**

Each data packet consisting of the object being sent (served) or received and its associated HTTP response header is wrapped in TCP/IP packets. The HTTP request is also wrapped in (usually) one TCP/IP packet. Each TCP/IP packet has a TCP/IP header. The number of bytes in the header represents the TCP/IP overhead for the TCP/IP packet. Since the overhead per TCP/IP packet is the size of its header, a 40-byte structure, multiply the number of packets by 40 to get the TCP/IP overhead.

#### **UDPOVERHEAD**

If the protocol for sending the data is UDP, then the UDP/IP overhead consists of only one component: the UDP/IP data-packet overhead. Each data packet of the object being served is wrapped in UDP/IP packets. The first UDP/IP packet has both an IP header of 20 bytes and a UDP header of 8 bytes. Subsequent packets have only an IP header of 20 bytes.

#### **Ack packet for HTTP client request**

An acknowledgment is an empty TCP/IP packet, i.e., a packet with a header but no data, used to signal the receipt of data or the receipt of a request for data. The overhead associated with sending or receiving each acknowledgment is 40-byte TCP/IP packet header. A server sends an acknowledgment when it receives a request for data (i.e., the object and its associated response header), and it receives an acknowledgment when it requests data. This request acknowledgment incurs an overhead of 40 bytes.

#### **Bytes of request**

Streaming request overhead is the number of bytes of the streaming request, which is estimated to be the length in bytes of the URL requested. The length of the URL can be determined from the logs. Each character in the URL is one byte, so the number of bytes of the streaming request is the number of characters in the URL.

#### **Size of stream**

### WMS/QuickTime

The size of the stream (or portion served) is extracted from the logs generated by the Akamai Windows Media or QuickTime streaming servers. The logs generally contain two relevant fields

- `sc_bytes`, which is the number of bytes the server reports as sending to the client
- `c_bytes`, which is the number of bytes the client reports as having received.

In the case where these two numbers do not agree, or one is not present in the logs, the higher of the two numbers is the value used for the size of the stream. It is important to note that these fields might not always agree because of server-player communication problems that might prevent the player from sending feedback to the server.

### Native Real Media

Native Real Media uses `sc_bytes` field to determine the size of the stream. This field is the only source of byte information for this stream format.

### Flash

Flash uses `sc-bytes` from the disconnect log line for billing purposes only, as this number contains additional bytes associated with rtmp connection establishment, maintenance, and bandwidth detection in the Flash client. However, because the disconnect action has no URL associated with it, Flash uses the `sc-stream-bytes` number for reporting. The `sc-stream-bytes` value is assigned per stream in a session and is associated with the stop action for that stream, making a URL available for reporting purposes.

## Number of packets

### WMS/QuickTime/Native Real Media

The number of packets equals the reported number of packets, as established in the Constants and Equations section. If the `s_pkts_sent` and `c_pkts_received` values are unreasonably high, then the number of packets used in the overhead calculation is the estimated number of packets, as established in the Constants and Equations section, calculated appropriately for the protocol.

### Flash

The number of packets equals the reported number of packets, as established in the Constants and Equations section. There are no alternate calculations and the Flash log has no packet data because the stream always uses the TCP protocol.

## Incoming ack packets

An acknowledgment is an empty TCP/IP packet, i.e., a packet with a header but no data, used to signal the receipt of data or the receipt of a request for data. The overhead associated with sending or receiving each acknowledgment is 40-byte TCP/IP packet header. A server sends acknowledgments when it receives data, and it receives acknowledgments when it sends data. In data transfers involving more than a handful of packets, one acknowledgment is usually generated for every two TCP/IP packets of data.

For example, to calculate the number of acknowledgments received by an edge server sending an object (and its associated response header) to an end user, you would first round the number of TCP/IP packets for sending the data (calculated above) to the nearest

highest even number. Then, you would divide this number by 2 to get the number of acknowledgments received. Multiplying by 40 yields the number of acknowledgment overhead bytes incurred when the end user's browser acknowledges to the edge server that the data was received.

**Incoming syn, ack, finack, and reset packets**

The opening and closing of a new connection between the end- user's browser and the edge server involves the sending of six empty TCP/IP packets. The TCP/IP new-connection overhead associated with sending each packet is its 40-byte header. Four of these packets are sent by the browser to the edge server (incoming syn, ack, finack, and reset packets), and the other two are sent by the edge server to the browser (outgoing synack and fin packets). The overhead involved in sending these packets is added only when the browser makes its first (or only) request to the edge server. Subsequent requests on the same connection are not charged this overhead.

**Outgoing synack and fin packets**

The opening and closing of a new connection between the end- user's browser and the edge server involves the sending of six empty TCP/IP packets. The TCP/IP new-connection overhead associated with sending each packet is its 40-byte header. Four of these packets are sent by the browser to the edge server (incoming syn, ack, finack, and reset packets), and the other two are sent by the edge server to the browser (outgoing synack and fin packets). The overhead involved in sending these packets is added only when the browser makes its first (or only) request to the edge server. Subsequent requests on the same connection are not charged this overhead.

**Outgoing TCP overhead packets**

Each data packet consisting of the object being sent (served) or received and its associated HTTP response header is wrapped in TCP/IP packets. The HTTP request is also wrapped in (usually) one TCP/IP packet. Each TCP/IP packet has a TCP/IP header. The number of bytes in the header represents the TCP/IP overhead for the TCP/IP packet. Consequently, to determine the overhead for all packets, first compute the number of TCP/IP packets required to send the data and the request (see "PACKETSIZE"). Since the overhead per TCP/IP packet is the size of its header, a 40-byte structure (TCPIPOVERHEAD), multiply the number of packets by 40 to get the total overhead.

## Total Bytes Calculations

For each log line with bytes that should be tabulated, protocol can be HTTP, UDP or TCP, and the overhead will be calculated per log line in a manner specific to each protocol. Flash is an exception in that it always uses the TCP protocol. The total bytes number is used for billing purposes.

**HTTP** Assuming each TCP packet carries `PACKETSIZE - TCPIPOVERHEAD` data bytes:

estimated outgoing bytes	= size of stream + ack packet for HTTP client request + outgoing TCP overhead packets + outgoing synack, and fin packets
estimated incoming bytes	= bytes of request + <code>DEFAULTHTTPOVERHEAD</code> + ack for HTTP client request + incoming ack packets + incoming syn, ack, finack, and reset packets
total bytes	= estimated outgoing bytes + estimated incoming bytes

**TCP** Assuming each TCP packet carries `PACKETSIZE - TCPIPOVERHEAD` data bytes:

estimated outgoing bytes	= size of stream + outgoing TCP overhead packets + outgoing synack, and fin packets
estimated incoming bytes	= bytes of request + incoming ack packets + incoming syn, ack, finack, and reset packets
total bytes	= estimated outgoing bytes + estimated incoming bytes

**UDP** Assuming each UDP packet carries `PACKETSIZE - IPOVERHEAD` data bytes, except the first packet which carries `PACKETSIZE - IPOVERHEAD - UDPOVERHEAD` data bytes:

estimated outgoing bytes	= size of stream + <code>UDPOVERHEAD</code> + ( <code>IPOVERHEAD</code> + <code>ETHERNETOVERHEAD</code> ) * number of packets;
estimated incoming bytes	= bytes of request;
total bytes	= estimated outgoing bytes + estimated incoming bytes



# Appendix B. Common Pitfalls in Report Comparisons

## In This Appendix:



- ◆ The following are some common errors or problems in interpreting reports. • 55
- ◆ LDS Log Data Different from Traffic Reports • 55
- ◆ 'Data not Final' Message in Contract Usage Reports • 55

The following are some common errors or problems in interpreting reports.

## Successful and Error Transactions in HTTP Reports

Content Delivery reports in Traffic Reports include data for successful HTTP transactions only, unless the user requests an explicit "errors" report. Successful transactions exclude those with HTTP status codes in the 400 and 500 ranges. Contract Usage reports include data for all status codes. Though error responses are generally a small part of traffic, this difference between Content Delivery and Contract Usage reports can cause confusion.

## LDS Log Data Different from Traffic Reports

The size of an object indicated in an LDS log line does not include any of the overheads described in "Detailed Overhead Calculations" on page 43. These overheads are generally not included in standard Web server or Streaming media server logs, and LDS follows that convention. In addition, traffic summary data is based on hits per second or megabytes per second; when those values are expanded, particularly over multiple days, the numbers are unlikely to exactly match.

## 'Data not Final' Message in Contract Usage Reports

The contract usage metric is the same measurement that appears on an Akamai invoice. The 'Data not Final' message in Contract Usage reports indicates that the final contract usage metric has not been officially computed for a particular month. While you can see a statistic for the last day of the month before it is marked final, there exists a slight chance that the number might change when the final numbers are run for the billing cycle. The final number is computed during the first week of the following month by Akamai's finance team.





# Appendix C. Response Codes

## In This Appendix:



- ◆ 000 Client-Side Abort • 57
- ◆ 100 Range – Informational Status Codes • 57
- ◆ 200 Range – Successful Status Codes • 57
- ◆ 300 Range – Redirection Status Codes • 58
- ◆ 400 Range - Client Errors Status Codes • 58
- ◆ 500 Range - Server Error Status Codes • 59
- ◆ 600 Range - Invalid Headers • 59

Descriptions of the standard status codes can be found in RFC 2616 “HTTP/1.1” (<http://www.w3c.org/Protocols> ).

## 000 Client-Side Abort

### 000 Client-Side Abort

The download was terminated by the end user, by hitting the Stop button, etc, before the Akamai edge server was even able to send back the response headers. This status code is defined by Akamai. Please note that LDS processing converts 200/206 response codes with error client abort into 000 in the log files.

## 100 Range – Informational Status Codes

100 – Continue. For Flash, response code 100 during a “connect-pending” event indicates “Waiting for application to authenticate.”

101 – Switching Protocols

## 200 Range – Successful Status Codes

Note that responses in the 200 ranges can occur for a HEAD request or client abort occurring after header transmission, in addition to the reasons listed below.

200 - OK. For Flash, response code 200 during a “connect,” “play,” “publish,” or “publish” events indicates “Successful.”

201 - Created

202 - Accepted

203 - Non-Authoritative Information

204 - No Content

205 - Reset Content

206 - Partial Content

Please note that LDS processing converts 200/206 response codes with error client abort into 000 in the log files.

## 300 Range – Redirection Status Codes

300 - Multiple Choices

301 - Moved Permanently

302 - Found. For Flash, response code 302 during a “connect” event indicates “Application is temporarily unavailable.”

303 - See Other

304 - Not Modified

305 - Use Proxy

307 - Temporary Redirect

## 400 Range - Client Errors Status Codes

400 – Bad Request. For Flash, response code 400 during a “connect” event indicates “Bad Request. Client connected to server using an unknown protocol.” For Flash, response code 400 during “play” or “publish” events indicate “Bad Request.”

401 – Unauthorized. For Flash, response code 400 during a “connect” event indicates “Connection was rejected by application script.” For Flash, response code 400 during “play” or “publish” events indicate “Access denied by application.”

402 – Payment Required

403 – Forbidden. For Flash, response code 400 during a “connect” event indicates “Connection was rejected by access module.” For Flash, response code 400 during a “play” event indicates “Play forbidden by stream module.”

404 – Not Found. For Flash, response code 400 during a “connect” event indicates “Application not found.” For Flash, response code 400 during a “play” event indicates “Stream not found.”

405 – Method Not Allowed

406 – Not Acceptable

407 – Proxy Authentication Required

408 – Request Timeout. Note: For streaming, 408 is considered a successful hit. For WMS, it implies that clients are disconnected from (or poorly connected to) the network and so the servers time them out. Please see “Streaming Hit” on page 25 for more information on WMS 408 responses. For Flash, response code 408 during a “stop” event indicates “Stream stopped because client disconnected.”

409 – Conflict. For Flash, response code 409 during a “connect” event indicates “Resource limit exceeded.” For Flash, response code 400 during a “publish” event indicates “Stream is already being published.”

410 – Gone

411 – Length Required

412 – Precondition Failed

413 – Request Entity Too Large. For Flash, response code 413 during a “connect” event indicates “License limit exceeded.”

414 – Request - URI Too long

415 – Unsupported Media Type. For Flash, response code 415 during a “play” or “publish” event indicates “Unsupported Media Type.”

416 – Requested Range Not Satisfiable

417 – Expectation Failed

## 500 Range - Server Error Status Codes

500 – Internal Server Error. For Flash, response code 500 during a “connect,” “play,” or “publish” event indicates “Internal server error.”

501 – Not Implemented

502 – Bad Gateway. For Flash, response code 502 during a “connect” event indicates “Bad gateway.”

503 – Service Unavailable. For Flash, response code 503 during a “connect” event indicates “Service Unavailable. (Too many connections pending for authorization by access module).”

504 – Gateway Timeout

505 – HTTP Version Not Supported

## 600 Range - Invalid Headers

600 Range response codes, Invalid Headers, are defined by Akamai.





## Appendix D. Geographic Regions

Traffic reports include the geographic region (country, state, or province) of the user requesting content. This data is generated by using the client IP address to look up the location in an EdgeScape database on the Akamai edge server.

EdgeScape files available on EdgeControl include the possible values for these regions. For more information, please go to [Support > Documentation > EdgeScape > Data Codes](#).



# Appendix E. Internet 500 Retail Categories

In This Appendix:



◆ Retail Categories • 63

The following Retail Categories are based on the Internet 500 for Retail in 2006. The Competitive Benchmark for Retail report uses these categories for industry comparisons.

## Retail Categories

- Apparel/Accessories
- Books/CDs/DVDs
- Computers/Electronics
- Flowers/Gifts
- Food/Drug
- Hardware/Home
- Health/Beauty
- Housewares/Home
- Jewelry
- Mass Merchant
- Office Supplies
- Specialty/Non-Apparel
- Sporting Goods
- Toys/Hobbies

