Fast Internet Content Delivery with FreeFlow

April 2000

Contents

Fast Internet Content Delivery with FreeFlow

1 Introduction: Overview of the system

Akamai's FreeFlow is a high-performance, fault-tolerant Web content delivery service. It works by directing requests for media-rich[1] Web objects away from busy home sites and to Akamai's network of over 2,750 servers located on over 150 networks in over 45 countries around the world. FreeFlow's sophisticated optimization algorithms use up-to-the-second information about Internet traffic conditions to map every end-user request to the Akamai server that will deliver the requested content most quickly. As a result, FreeFlow provides the following benefits to Web businesses: faster downloads for end-users, reduced load on the home site, flash crowd protection, easier Web site management and infrastructure scaling, and the ability to distribute media-rich objects effectively. Important additional features, such as fault-tolerance in the event of server or network failures and quick response to bad Internet traffic conditions, ensure that end-users of FreeFlow-enabled Web sites experience a consistently high quality of service regardless of prevailing network conditions.

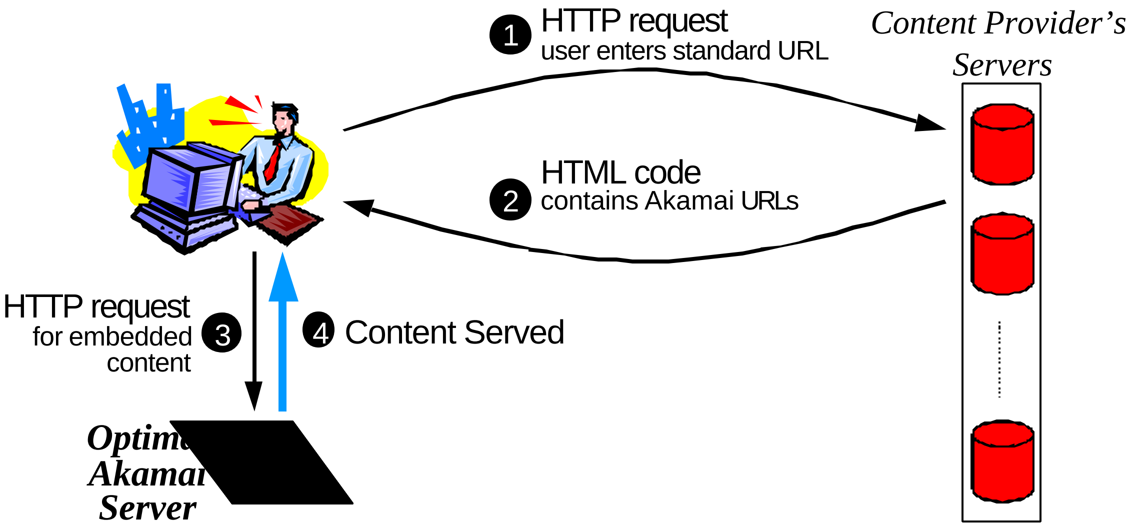


Figure 1. Typical user interaction with an Akamaized web site.

Figure 1 illustrates a typical user interaction with a FreeFlow-enabled Web site. First, the user's browser sends a request for a Web page to the site. In response, the Web site returns the appropriate HTML code as usual, the only difference being that the enclosed embedded object URLs have been modified to point to the Akamai network. As a result, the browser next requests and obtains the media-rich embedded objects from an optimally located Akamai server, instead of from the home site. In the event of new content, the Akamai server first retrieves the object from the home site and then caches it. Since 70-90% of Web traffic consists of downloads of media-rich embedded objects such as images, this approach dramatically reduces the traffic burden on the home server. Also, since content is delivered by easy-to-reach servers, download time decreases.

[1] Media-rich Web objects include: embedded objects such as images and scripts, and large file downloads such as audio and video files and program executables.

The technical benefits of FreeFlow allow Web businesses to focus on core business issues such as marketing, product fulfillment, and content development, instead of on the complex and often intractable technical obstacles to delivering content across the Internet in an efficient and effective way. Through the combination of new technology and an aggressive network deployment strategy, Akamai Technologies enables Web businesses to serve content from a global network of servers, overcoming the limitations of today's restrictive model for delivering Web content to end users. No longer will "flash-crowds" – sudden traffic bursts that can render a Web site or an entire network effectively unreachable – make Web sites victims of their own success. Akamai offers the first real, cost-effective solution.

This document presents the technology behind FreeFlow that makes all of these benefits possible. The next two sections describe the scope and composition of the Akamai network and explain how Web requests get directed to this network. The core FreeFlow technology, consisting of the algorithms that help direct each request to the optimal Akamai server, is explained in section 4. Availability and fault-tolerance are covered in section 5. Sections 6-8 describe server operation, FreeFlow monitoring capabilities, and other service offerings.

2 The Akamai Network

Akamai has deployed the world's largest fault-tolerant network dedicated to the delivery of Web content. FreeFlow enables content providers to utilize this network for the distribution of the company or organization's content across the Internet. Akamai deploys its servers in data centers and Points of Presence (POPs) of major Internet and communications carriers to achieve network diversity, fault-tolerance, and network proximity to users. Deployed internationally, the FreeFlow network gives businesses an instant, highly cost-effective global presence.

Currently, more than 2,750 Akamai FreeFlow servers are deployed at more than 150 network providers around the world. This installation includes global providers with data centers in Europe, the Pacific Rim, Australia, Asia, Africa and South America, putting Akamai FreeFlow servers in over 45 different countries. With more than 15 Gbps of egress capacity, the FreeFlow service network is adequately provisioned to handle a significant percentage of current and future Web traffic.

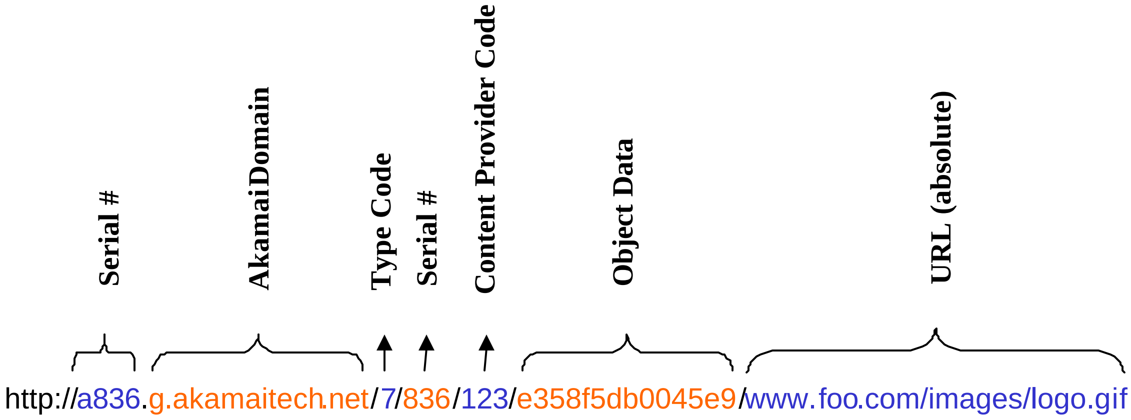
3 ARLs and AKAMAIZER: Directing traffic to the Akamai network

FreeFlow **Akamaizer** is the tool that tags embedded Web objects for delivery via the FreeFlow network, transforming ("akamaizing") their URLs into Akamai Resource Locators (**ARL**s). ARLs contain a number of fields that aid in the content delivery process. Their format is best described with an example:

A typical embedded object URL such as

http://www.foo.com/images/logo.gif

would be transformed into the following ARL:



The ***serial number*** identifies a virtual "bucket" of content – a group of akamaized Web objects that will always be served from the same set of Akamai servers. Grouping content using serial numbers allows FreeFlow to balance load among servers and efficiently use storage resources such as server RAM. Serial number values range from 0 to 2047.

The ***Akamai domain*** ensures that requests for akamaized content travel directly from the user to an Akamai server, completely avoiding the object's home site. With rare exception, this field will be set to *g.akamai.net*.

The ***type code*** defines the way in which the FreeFlow content servers interpret an ARL. It is used to define and select extensions to the standard Freeflow service. Currently supported functionality includes support for object expiration times, object fingerprint checking, and choices for query string interpretation. See Appendix A for a complete list of type code definitions.

The ***content provider code*** uniquely identifies an Akamai customer (content provider) account. It is used for usage reports, logging, and billing purposes. Customers may be assigned multiple codes to employ for special events, billing against cost centers, or for tracking usage of specific objects or content areas of their sites.

The ***object data*** field is used to guarantee object freshness. Depending on the typecode in use, this field will contain either the object's expiration time, or a string that uniquely identifies a particular version of the object. In the latter case, when the object is modified, its object data field changes, so its ARL (embedded in the container HTML page) changes as well. Any user that requests the page after an update receives a page that points to the new version of the object. The old version is never referenced and cannot be mistakenly returned in place of the more up-to-date information. Therefore, the object data ensures that FreeFlow will never serve a stale object. This field may be set up to be the cryptographic hash (MD5) of the object contents, or simply a version number.

The original ***absolute URL*** is used by FreeFlow content servers to retrieve the object from the content provider's home site the first time the object is requested. For security purposes, the Web server hostname in the original URL must be registered with Akamai in order for the ARL request to be honored.

Akamaizing a Web site involves transforming targeted URL tags in HTML pages into ARLs. The Akamaizer utility automates this task. Akamaizer can be run on individual URLs, HTML files or Web site document trees. It scans through HTML and targets the following objects to akamaize:

- `SCRIPT' tags looking for a `src' attribute.
- `EMBED' tags looking for a `src' attribute.
- `INPUT' tags looking for a `src' attribute.
- `IMG' tags looking for a `src' attribute.
- `BODY' tags looking for a `background' attribute.
- `A' tags looking for a `href' attribute with an associated value that ends in one of `.mov', `.wav', or `.avi'.

Akamaizer can be integrated into many different publishing environments, and is typically run in the staging phase, right before content gets pushed out to the live production environment. It can also be easily integrated into dynamic publishing environments. Akamaizer currently exists in the following formats: Perl, C++, Java, COM. It has also been integrated into both the Vignette StoryServer and Blue Martini publishing systems. Integration with ATG's Dynamo will follow shortly. Because each customer's needs are different, Akamai provides initial and ongoing support for Akamaizer planning and integration.

4 FreeFlow DNS: Directing requests to the best Akamai server

All user requests for ARLs are directed to the Akamai network by the server domain field (set to *g.akamai.net*) in each ARL. The FreeFlow DNS system ensures fast delivery of the requested content by resolving each **.g.akamai.net* server name to the IP address of the Akamai server that will deliver the requested content to the user most quickly. This resolution is different from conventional DNS name resolution in that the response is based not only on the server name, but also on the source address of the DNS query and current Internet traffic conditions.

FreeFlow DNS is implemented as a 2-level hierarchy of DNS Web servers: 50 high-level *.akamai.net* servers (HLDNS) and 2000 low-level *.g.akamai.net* servers (LLDNS). Each HLDNS server is responsible for directing each query it receives to a LLDNS server that is close to the requesting client. The LLDNS servers perform the final resolution of IP name to server address, directing each client to the Akamai server that is optimally located to serve the client's requests. Because FreeFlow DNS continuously monitors network conditions and the status of each server, it can respond to network events within several seconds.

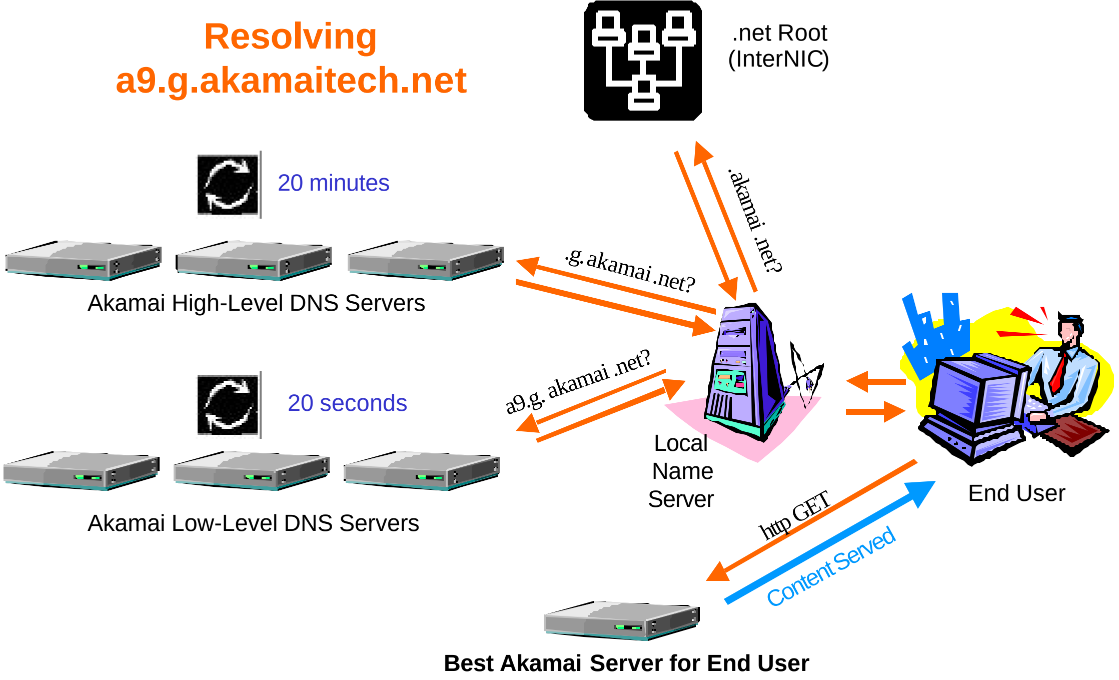


Figure 2. Resolving a9.g.akamai.net

Figure 2 illustrates how DNS queries travel through this hierarchy. When a browser makes a request for an ARL, it first contacts its local DNS server, asking it to resolve the *a9.g.akamai.net* server name. In the absence of a cached response, it does so using iterative DNS queries. The local server first contacts a *.net* root server, which responds with a list of Akamai HLDNS servers. When the local DNS server contacts one of these, it receives a list of LLDNS servers that are close to it. It then contacts one of the LLDNS servers, which responds with the IP address of the optimal content server for this request. The local DNS server returns this IP address to the browser, which then fetches the content from that machine.

Of course, just as in conventional DNS name resolution, FreeFlow DNS enables caching of DNS responses in order to avoid having every request incur the delay of three levels of DNS queries before content can be downloaded. The TTLs of the responses are set in such a way as to balance the benefits of caching with the chief goal of FreeFlow DNS: keeping the client-to-server mapping up to date with current Internet traffic conditions. Since the responses obtained from the root *.net* servers do not vary with network conditions, they have a TTL of two days. The responses returned by HLDNS servers are based on a network map that is recomputed every 7-10 minutes, so these responses have a TTL of 20 minutes. Since LLDNS servers generate name resolutions based on maps that are recomputed every 2-10 seconds, the TTL of LLDNS responses is 20 seconds. Although a TTL of 20 seconds seems very low, it does not negatively impact DNS resolution performance. First, the LLDNS contacted is always very close (in a network sense) to the user's local DNS server, so their communication is very fast. Second, the resolution is cached long enough to be shared among a number of clients. Most importantly, this low TTL allows FreeFlow to almost instantaneously respond to changing network conditions, ensuring that each request is served from the optimal Akamai server.

5 High Availability via Fault-tolerance

In addition to good performance, FreeFlow also provides Web businesses with a **highly available** content delivery service. It can do this because the FreeFlow network is designed to be highly fault-tolerant, with no single point of failure. This section describes the three types of service-denying failures that can occur in today's hosting model, and explains how FreeFlow maintains full service operation in the face of each type of failure.

The three types of service-denying failure are:

a. Machine Failures—hardware or software components responsible for serving content fail. In a Web hosting environment, resolution may require no more than a system reboot taking minutes or a system rebuild including data restoration from tape lasting several days.

b. Network Outages—router or switch hardware failures, operations errors, fiber cuts, over-capacity failures, and denial-of-service attacks can all cause networks to fail in delivering traffic. Since network problems are usually difficult to diagnose, resolution is often time consuming, taking hours or days. Short-term solutions often lead to prolonged periods of highly degraded service.

c. Data Center Outages—power failures, fires, operations errors, and major events such as earthquakes can all lead to outages at data centers. As with network outages, resolution of data center outages can be time-consuming.

The FreeFlow network remains fully operational in the face of all three types of failure.

5.1 Machine Failures

To avoid service denial in the case of a failure of a content server machine, FreeFlow employs the *Buddy System*. Each server is assigned a companion/monitor system termed a buddy. Buddies monitor each other with heartbeat messages sent over a backend private network connection.. If a server fails, its buddy takes over for it by assuming the failed system's IP address via ARP[2]. Failover time is 1 second.

FreeFlow is also fault-tolerant in the case that an Akamai DNS server (HLDNS or LLDNS) fails. Just as in conventional DNS, Akamai DNS servers return a list of addresses in response to each query. If one server is down, the user will still be able to contact another one on the list.

[2] The Address Resolution Protocol (ARP) translates IP addresses into Ethernet addresses.

5.2 Network and Data Center Outages

FreeFlow network monitors (described in 4.1) continuously monitor the availability and capacity of host networks and FreeFlow servers. In the case of a network or data center outage, the communication cost associated with that network or data center is set to infinity, causing the mapping algorithms to eliminate the data center or network from both the high-level and fast maps. In other words, no FreeFlow traffic will be sent through the failing network or data center. Failover time is only 1-2 minutes, since the FreeFlow DNS maps are updated very frequently.

5.3 Content Provider Must Provide Robust Home Site

Although the FreeFlow network is fault tolerant, the content provider still must ensure that the home Web site is sufficiently robust and fault tolerant to handle normal operation. Since the home site must handle all initial requests to the site, if the site goes down, no requests will reach the FreeFlow network. An advantage of the FreeFlow solution is that content provider resources previously allocated to provisioning additional hardware and bandwidth to overcome performance and distribution problems can be reallocated to other needs, such as additional server and RAID processors, or other fault-tolerance solutions.

6 FreeFlow Content Servers

FreeFlow content servers are PentiumII-based caching appliances with 1 GB of RAM and 40-80 GB of disk storage. They run the Linux operating system with a modified kernel tuned to provide excellent network performance. Each machine is capable of saturating a 100 Mbps Ethernet segment.

The caching engine running on the servers is proprietary Akamai software. It provides important HTTP 1.1 functionality such as byte-range transfers. In response to *If-Modified-Since* requests, FreeFlow caches always answer *Not-Modified*, without having to check with the home site. They can do this because the ARL object data field (described in section 3) guarantees freshness. The caching engine honors and forwards all HTTP *Expires* headers.

7 Network and Performance Monitoring

Akamai strives to keep content providers informed as to the performance of their sites and the FreeFlow network in general. To that end, Akamai provides several different types of network and performance monitoring capabilities.

7.1 Network Operations Center

Akamai staffs its NOC 24x7x365 to respond to any problem that may arise on the FreeFlow network. All systems on the FreeFlow network are monitored to ensure that key processes are running, systems have not exceeded capacity, and regions are interacting properly.

The FreeFlow service reduces the need for standard human intervention in network problems because of the fault-tolerance of the system. When systems go down, nearby systems take over automatically and the Akamai NOC staff members take immediate remedial action, following well-defined escalation procedures. During special events, Akamai remains in constant contact with the customer, providing feedback and reporting on system performance throughout.

7.2 Akamai Reporting Services

Akamai provides each FreeFlow content provider with access to Akamai reporting services providing them with site usage statistics. The services include both real-time and historic analysis of customer data. Akamai Traffic Analyzer, the real-time component of these services, has multiple monitoring views that enable quick access to network and customer-specific traffic.

Akamai Reporter complements Traffic Analyzer by extending its reporting capabilities to allow for viewing of historical data. Reporter culls its information from existing FreeFlow log files to provide for exploration of site traffic patterns in the data. Reporter also provides customers with the option to export data to other applications for more detailed offline analysis.

8 Additional Service Offerings

In addition to providing Web businesses with the ability to speed up object downloads, Akamai also offers the fast and reliable delivery of video and audio streams. The *FreeFlow Streaming* service supports Apple QuickTime4 and Microsoft Windows Media streaming formats, with RealG2 support to come soon.

Akamai's *FirstFlow* service helps Web businesses deploy fast, highly available home sites. It uses the same monitoring and mapping technology found in FreeFlow to intelligently distribute requests among all of the servers of a mirrored Web site.

The Akamai Accelerated Network Program provides Internet Service Providers with a zero-cost method of reducing bandwidth costs and improving customer satisfaction. By locating FreeFlow servers in their POPs, AAN participants reduce the amount of bandwidth needed to deliver Web content to their subscribers, and decrease download time as well. Because each server is managed by Akamai, AAN provides all of the benefits of deploying a Web cache without the cost of operating it.

9 Fast Content Delivery With FreeFlow

In summary, FreeFlow provides high performance Internet content delivery by directing requests for media-rich Web objects to the Akamai network. Content is tagged for FreeFlow delivery by the Akamaizer script, which converts URLs to ARLs. Akamai's proprietary DNS system ensures that each request for an ARL is directed to the Akamai server that will service the request the most quickly. By creating request-to-server mappings that are based on up-to-the-second information on current Internet traffic conditions, FreeFlow routes end-user requests around network problem areas and delivers content to users in the fastest, most efficient way possible.

Appendix A: Typecodes

Typecode 7

The object data field of the ARL contains a string of characters that uniquely identifies a particular version of the object. This can be an MD5 hash of the object, a version number, or any other data that changes when the object changes. Upon receiving a request for an object, an Akamai server checks to see if the object data field in the ARL being requested matches that of the corresponding object in its cache (if any). If it matches, then the object is served right away. If it doesn't match, the server fetches the new version of the object from the home server before serving it to the user.

Example: http://a9.g.akamai.net/7/9/21/aaa7a80f016a2c/www.foo.com/images/logo.gif

Typecode 6

The object has a TTL of 30 minutes. This means that when a request is received for an object that has been in the cache for 30 minutes or longer, an If-Modified-Since query is sent to the object's home server to check freshness, and a new version is retrieved if necessary. The object is then served to the user, and the TTL clock is reset.

Example: http://a9.g.akamai.net/6/9/21/000/www.foo.com/images/logo.gif

Typecode f

The object TTL, expressed as a number of seconds, minutes, hours or days is specified in the ARL.

Example: http://a9.g.akamai.net/f/9/21/5h/www.foo.com/images/logo.gif

Typecode n

The Akamai content servers will prune query strings found at the end of ARLs.

Content providers often append a question mark ("?") to a URL followed by some data, such as inputs entered by the user, information about the page from which the request was generated, or other tracking or control information. The default behavior for a ghost is to cache an object using its ARL, including any appended query strings, as a key. This is the desired behavior in most situations, but causes performance problems when content providers append many different query strings to the ARLs of the same object. The object is cached multiple times in each cache, causing the Akamai content server to retrieve the object from the home server on every user request.

Typecode n solves this problem by instructing the Akamai server to ignore everything that appears after the "?," resulting in each object being retrieved from the home server only once.

The information that is pruned can be retrieved from the Akamai logs.

Example:
http://a9.g.akamai.net/n/9/21/aaa7a80f016a2c/www.foo.com/images/logo.gif?123456663

Typecode v

This typecode combines the functionality of typecodes n and f. Query strings are pruned, and a TTL is used to ensure object freshness.

Example: http://a9.g.akamai.net/v/9/21/5h/www.foo.com/images/logo.gif?1234452345