Network Caching

Networking Challenges with Internet and Intranet Growth

Internet and intranet traffic is growing at a phenomenal rate—doubling every 100 days. Such rapid increase in network traffic has created numerous networking challenges for Internet service providers (ISPs) and enterprises, including:

- WAN bandwidth congestion and high transmission costs
- Maximizing network service quality
- Maximizing and controlling the availability of Internet and intranet content as seen by clients
- Cost-efficient network scalability

Solution—Localize Traffic Patterns

The most efficient solution to these networking problems is to use your existing network infrastructure to localize traffic patterns, enabling content requests to be fulfilled locally. This solution addresses the preceding networking challenges in the following ways:

- Accelerated content delivery—Traffic localization accelerates content delivery by locally fulfilling content requests rather than traversing the Internet and intranet to a distant server farm. This solution helps to protect your network from uncontrollable bottlenecks, delivering more consistent network service quality and content availability.
- Optimized WAN bandwidth usage—Traffic localization minimizes redundant network traffic that traverses WAN links. As a result, WAN bandwidth costs either decrease or grow less quickly. This bandwidth optimization increases network capacity for additional users/traffic and for new services such as voice.

Traffic localization is a traffic engineering problem because it requires network intelligence to optimize traffic flows based on specified parameters. Therefore, the first step in building a traffic localization solution is to ensure that your existing network supports this capability. This capability can be achieved by enabling transparent redirection technology, such as the Web Cache Communication Protocol (WCCP), at key points within your network.

Once the right network foundation is in place, network caches are added into strategic points within your existing network to complete the traffic localization solution. Network caches transparently cache or store frequently accessed content and then locally fulfill successive requests for the same content, eliminating repetitive transmission of identical content over WAN links.

This is how the two components work together to localize traffic patterns:

1. A user requests a Web page from a browser.
2. The network analyzes the request, and based on certain parameters, transparently redirects it to a local network cache.
3. If the cache does not have the Web page, it will make its own Web request to the original Web server.
4. The original Web server delivers the content to the cache, which delivers the content to the client while saving the content in its local storage. That content is now cached.
5. Later, another user requests the same Web page, and the network analyzes this request, and based on certain parameters, transparently redirects it to the local network cache.
6. Instead of sending the request over the Internet and intranet, the network cache locally fulfills the request. This process accelerates the delivery of content and reduces WAN bandwidth usage.

Cisco Content Engines Caching Service

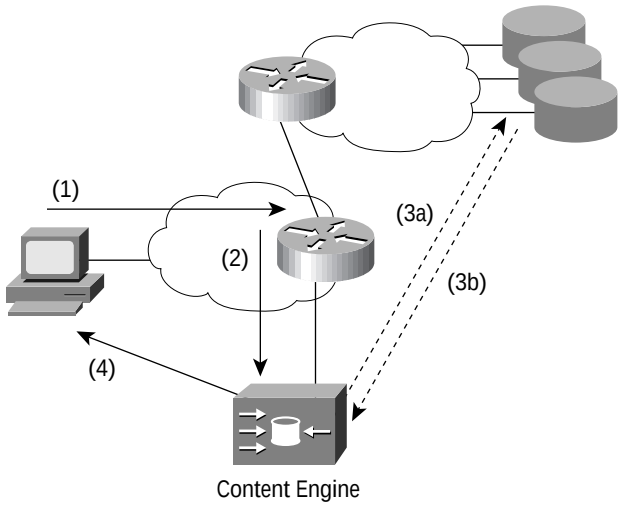
The Cisco network caching solution is comprised of the Cisco Content Engine caching service working in conjunction with your existing network infrastructure. Cisco Content Engines are content services platforms that accelerate content delivery, ensuring maximum scalability and availability of content. The following sections provide in-depth technical information on the Cisco network caching solution.

Transparent Network Caching

A content engine transparently caches as follows (Figure 1):

1. A user requests a Web page from a browser.
2. The WCCP-enabled router analyzes the request, and based on TCP port number, determines if it should transparently redirect it to a content engine. Access lists can be applied to control which requests are redirected.
3. If a content engine does not have the requested content, it sets up a separate TCP connection to the end server to retrieve the content. The content returns to, and is stored on, the content engine.
4. The content engine sends the content to the client. Upon subsequent requests for the same content, the content engine transparently fulfills the requests from its local storage.

Figure 1 Transparent Network Caching



Because the WCCP router redirects packets destined for Web servers to a content engine, the content engine operates transparently to clients. Clients do not need to configure their browsers to point to a specific proxy cache. This is a compelling feature for ISPs and large enterprises, for whom uniform browser configuration is expensive and difficult to manage. In addition, the content engine operation is transparent to the network—the router operates entirely in its normal role for nonredirected traffic.

Hierarchical Deployment

Because a Cisco Content Engine is transparent to the client and to network operation, customers can easily place content engines in several network locations in a hierarchical fashion. For example, if an ISP deploys a content engine at its main point of access to the Internet, all of its points of presence (POPs) benefit (Figure 2). Client requests hit the Cisco Engine and are fulfilled from its storage. To further improve service to quality clients, ISPs can deploy content engines at each POP. Then, when a client accesses the Internet, the request is first redirected to the POP content engine. If the POP content engine is unable to fulfill the request from local storage, it makes a normal Web request to the end server. Upstream, this request is redirected to the Cisco Content Engine at the main Internet access point. If the request is fulfilled by the Cisco Content Engine, traffic on the main Internet access link is avoided, the origin Web servers experience lower demand, and the client experiences better network response times. Enterprise networks can apply this hierarchical-transparent architecture to benefit in the same way (Figure 3).

Figure 2 Hierarchical Implementation of Content Engines (ISP)

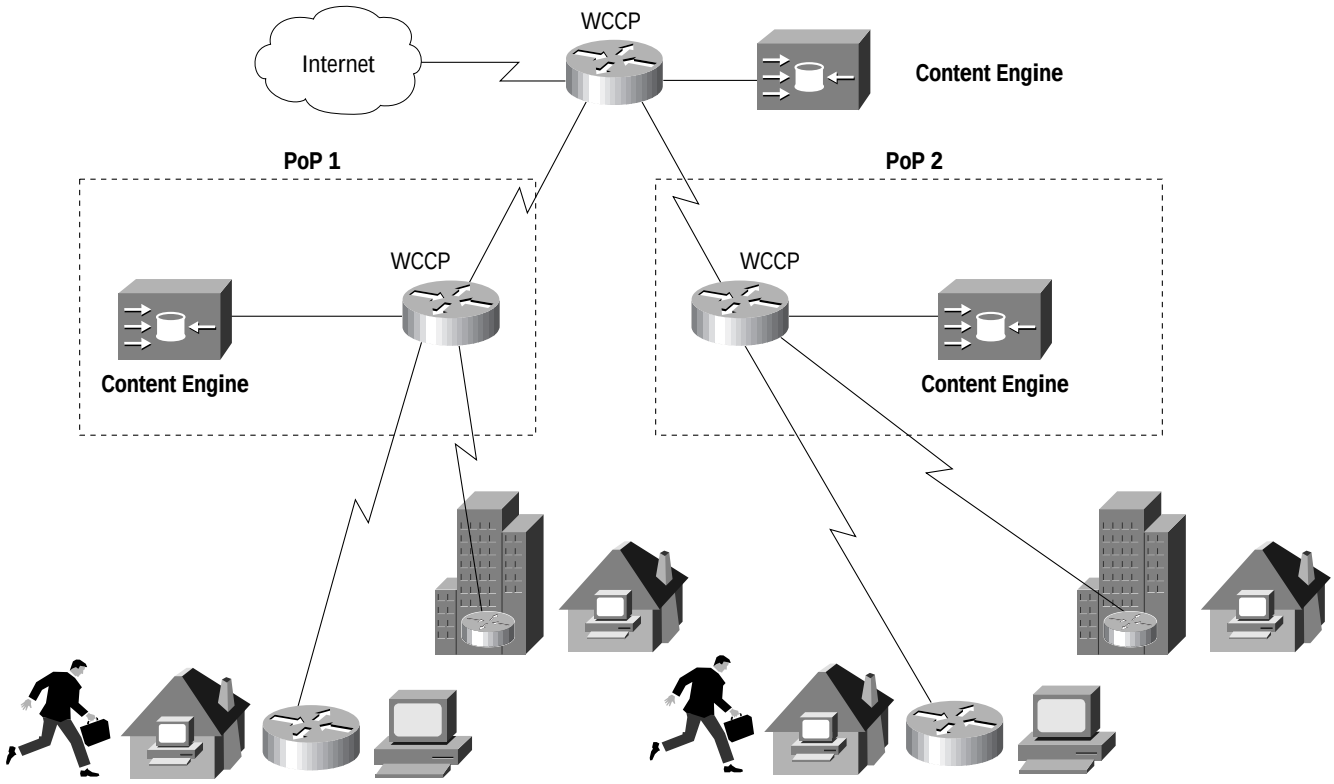
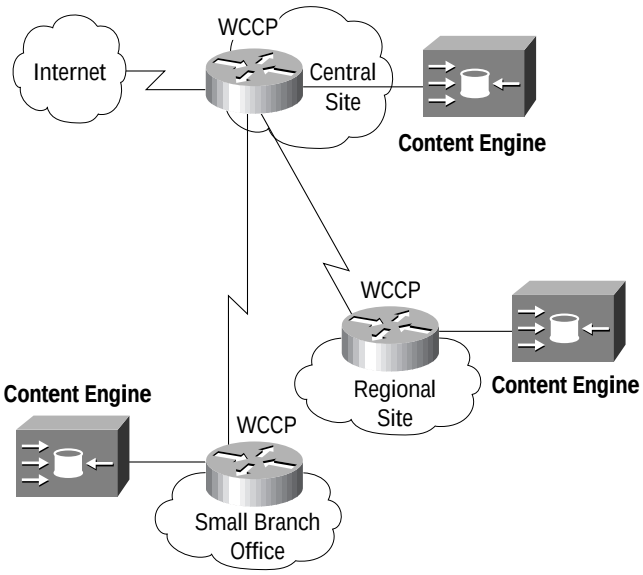


Figure 3 Hierarchical Implementation of Content Engines (Enterprise)



Scalable Clustering

The Cisco caching solution was architected to enable network administrators to easily cluster content engines to scale high traffic loads. This design approach allows customers to linearly scale performance and storage as content engines are added. For example, a single Cisco Content Engine 7320 can support more than 155 Mbps of traffic and 396 GB of storage; adding a second Cisco Content Engine 7320 provides support for more than 310 Mbps throughput and 792 GB of storage. Up to 32 content engines can be WCCP clustered together.

This linear scalability is achieved because of the manner in which WCCP-enabled routers redirect traffic to content engines. WCCP-enabled routers perform a hashing function on the incoming request's destination IP address, mapping the request into one of 256 discrete "buckets." Statistically, this hashing function distributes incoming requests evenly across all buckets. In addition, these buckets are evenly allocated among all content engines in a cluster. WCCP-enabled routers ensure that a certain content engine deterministically fulfills requests for a certain destination IP address on the Internet. Empirically, this distribution algorithm has consistently demonstrated even load distribution across a content engine cluster. Most of the popular Web sites have multiple IP addresses, thus preventing uneven load distribution.

When the customer adds a new content engine to the cluster, the WCCP-enabled router detects the presence of the new content engine and reallocates the 256 buckets to accommodate the additional content engine. For example, the simplest installation using one router and one content engine assigns all 256 buckets to the single content engine. If a customer adds another content engine, the WCCP-enabled router redirects packets to the two content engines evenly—128 buckets are allocated to each content engine. If the customer adds a third content engine, the WCCP-enabled router assigns 85 or 86 buckets to each of the three content engines.

Customers can hot-insert content engines into a fully operating cache cluster. In this situation, the WCCP-enabled router automatically reallocates the buckets evenly among all cache cluster members, including the new content engine. Because a new content engine will not have any content, it will incur frequent cache misses until enough content has been populated in its local storage. To alleviate this cold startup problem, the new content engine, for an initial period, sends a message to the other cache cluster members to see if they have the requested content. If they have the content, they will send it to the new content engine. Once the new content engine determines it has retrieved enough content from its peers (based on configurable numbers), it will handle cache misses by directly requesting the content from the end server rather than from its peers.

Fault Tolerance and Fail Safety

If any content engine in a cluster fails, the cluster automatically "heals" itself. The WCCP-enabled router redistributes the failed content engine's load evenly among the remaining content engines. The cluster continues operation using one less content engine, but operation is otherwise unaffected.

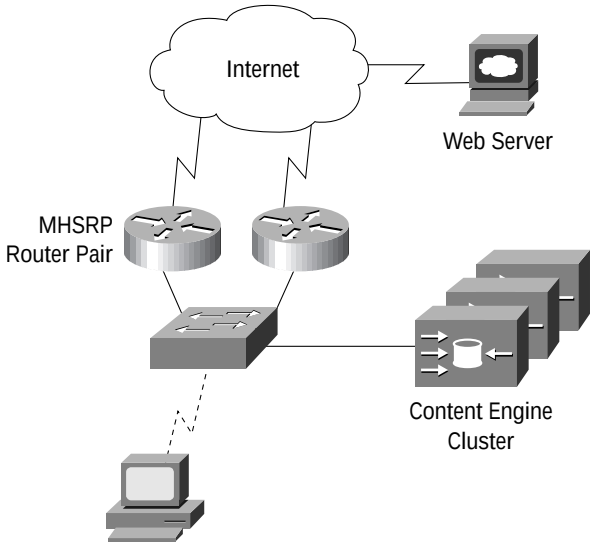
The Cisco network caching solution enables a WCCP-enabled, Multigroup Hot-Standby Router Protocol (MHSRP) router pair to share a content engine cluster, creating a fully redundant caching system. This is referred to as "WCCP multihoming." If the WCCP-enabled router fails, existing Cisco IOS fault tolerance and fail-safe mechanisms are applied. For example, a hot-standby router could dynamically take over operations, redirecting Web requests to the cluster.

If an entire content engine cluster fails, the WCCP-enabled router automatically stops redirecting traffic to the content engine cluster, sending clients' Web requests to the actual destination Web site in the traditional fashion. This loss of the entire cluster can appear to users as an increase in download time for Web content, but has no other significant effect. This designed-in, failsafe response is made possible because cluster is not directly in line with clients' other network traffic.

WCCP Multihome Router Support

As previously mentioned, the Cisco network caching solution enables a content engine cluster to "home" to multiple WCCP-enabled routers for added redundancy. Thus, Web traffic from all of the WCCP home routers will be redirected to the cluster. For example, a content engine cluster that is homing to both routers in a MHSRP router pair creates a fully redundant caching system, eliminating any single points of failure (Figure 4).

Figure 4 Fully Redundant Content Engine Cluster Configuration



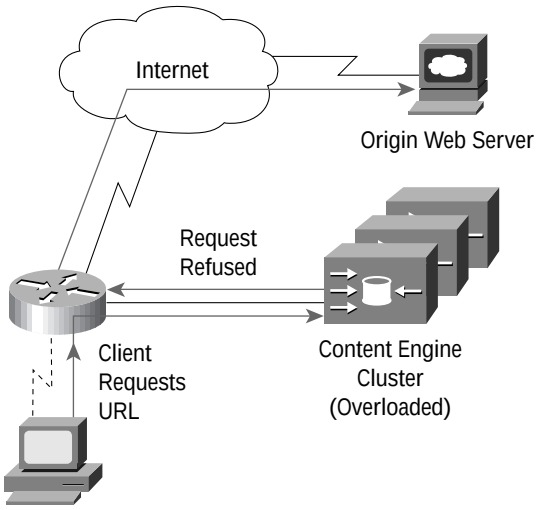
Overload Bypass

With a sudden Web traffic surge, a content engine cluster could become overloaded. To gracefully handle this overload situation, each content engine detects when it is overloaded, refuses additional requests, and forwards them to the origin Web servers. The origin Web servers respond directly to the clients because the bypassed requests were not handled by a content engine (Figure 5).

The overloaded content engine will resume accepting requests when it determines that it has the resources to do so without retriggering overload bypass in the near future. The overload bypass on/off triggers are automatically determined by CPU and file system load. In the extreme situation that the content engine becomes so overloaded that it is unable to respond to the basic WCCP status check messages from its home router, the WCCP home router will remove the content engine from the cluster and reallocate its buckets.

Thus, overload bypass ensures that a content engine cluster does not introduce abnormal latencies and maintains network availability even under unusually high traffic conditions.

Figure 5 Overload Bypass



Dynamic Client Bypass

Some Web sites require clients to be authenticated using the client's IP address. However, when a network cache is inserted between a client and a Web server, the Web server only sees the cache's IP address and not the client's IP address.

To overcome this issue and similar situations, the Cisco Content Engine has a dynamic client bypass feature that effectively allows clients, under certain conditions, to bypass content engines and directly connect to origin Web servers. The result is that a Cisco Content Engine can preserve existing source IP authentication models and pass through server error messages to clients. Because the content engine dynamically adapts to these situations, less management is required to ensure content engine transparency.

In Figure 6, a client issues a Web request, which is redirected to a content engine. If the content engine does not have the content, it will try to fetch the content from the origin Web server.

In Figure 7, if the server responds to the content engine with certain HTTP error return codes (such as 401-Unauthorized request, 403-Forbidden, or 503-Service Unavailable), the content engine will invoke the dynamic client bypass feature. The content engine will dynamically store a client IP-destination IP address bypass pair, so that future packets with this IP address pair will bypass the content engine. The content engine sends an automatic HTTP retry message to the client's browser.

In Figure 8, when the client's browser automatically issues a reload, the request will be redirected to the content engine. However, when the bypass table is checked and the request matches one of the table entries, the content engine will refuse the request and send it directly to the origin Web server. Thus, the origin Web server will see the client's IP address, authenticate the client, and respond directly to the client.

Figure 6 Dynamic Client Bypass

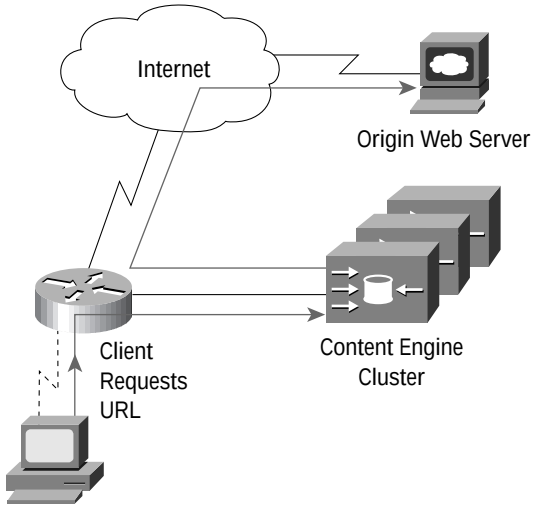


Figure 7 Dynamic Client Bypass

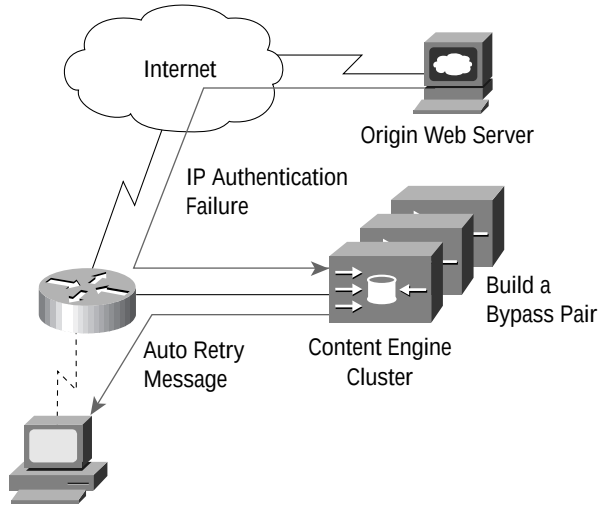
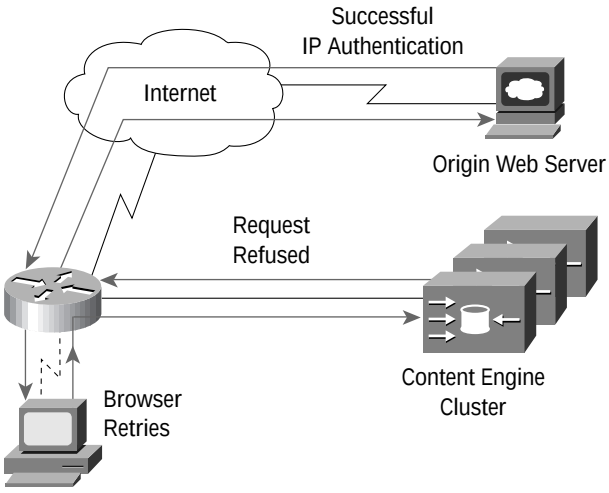


Figure 8 Dynamic Client Bypass



Reverse Proxy Caching

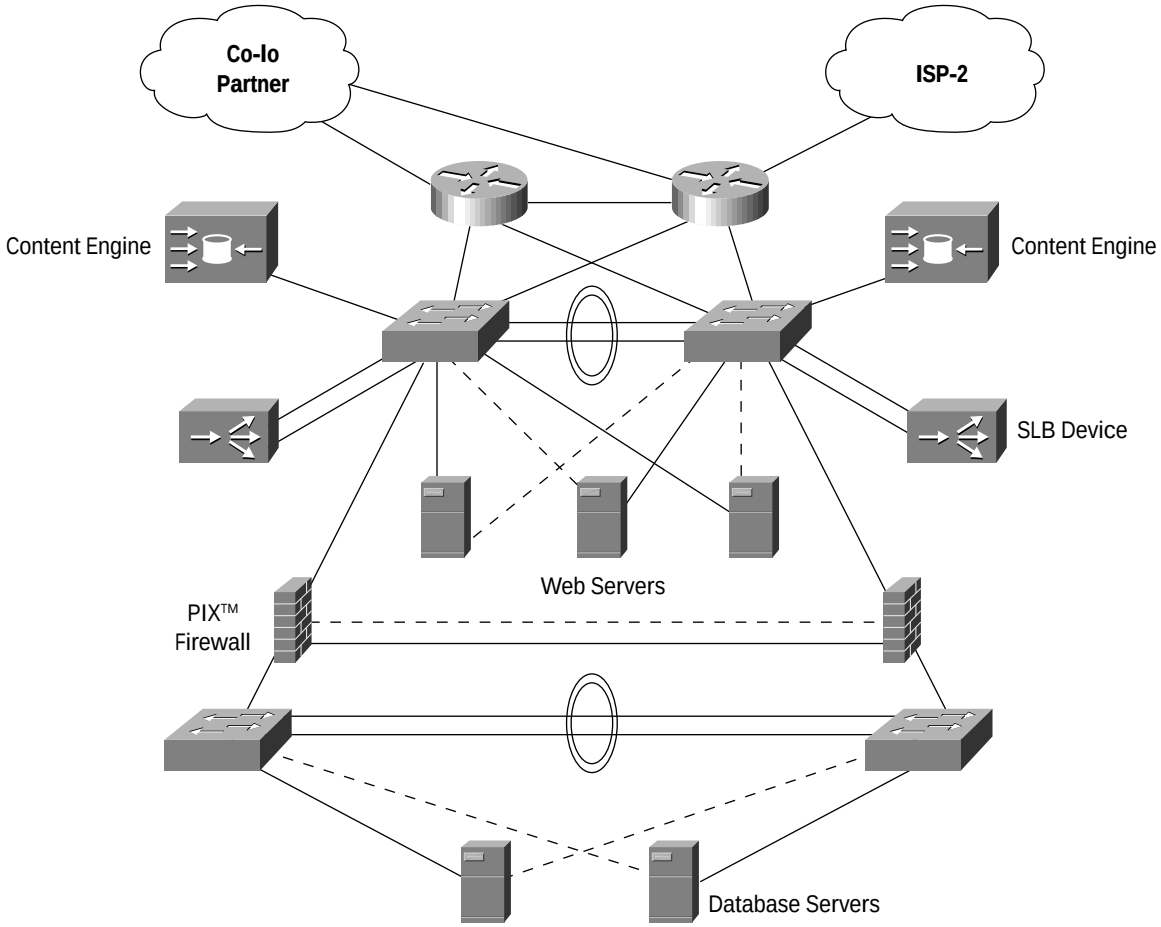
Content engines are frequently deployed nearby clients to ensure faster network response time and minimal WAN bandwidth usage. Thus, the content engines are caching the clients' most frequently accessed content. In addition, content engines can also be deployed in front of Web server farms to increase the server farm capacity and improve Web site performance. This configuration is called reverse proxy caching because the content engines are only caching content from the servers for which they are acting as a front-end.

This feature is particularly important when content engines are acting as front-ends for server farms in which certain content is dramatically more popular than other content on the servers. Using reverse-proxy caching allows administrators to prevent a small number high-demand URLs from impacting overall server performance. Better yet, this means the high-demand URLs do not have to be identified, manually replicated, or independently managed from the bulk of the URLs on the servers.

Reverse Proxy Caching Function

In Figure 9, each content engine "homes" to WCCP-enabled routers/switches that are supporting server farms. When an incoming Web request reaches a WCCP-enabled router/switch, the router/switch performs a hashing function on the incoming request's source IP address and port number, mapping the request into one of 256 discrete "buckets." Statistically, this hashing function distributes incoming requests evenly across all buckets. In addition, these buckets are evenly allocated among all content engines in a cluster.

Figure 9 Reverse Proxy Caching



Because the hashing function is based on source IP address and port number instead of destination IP address, a given Web object could be stored in multiple content engines in a cluster. By spreading popular content across a cache cluster, reverse proxy caching allows multiple content engines to service requests for very popular content. Thus, additional content engines can be added to a cluster to incrementally scale the performance of a popular site and decrease content download latency.

Note that the reverse-proxy caching could also be done by hashing on a destination IP address. But in this case, all requests would have the same destination IP address and would be redirected to one content engine. If you do not need to scale beyond one content engine as a front-end to a server farm, then this method is sufficient.

Ensuring Fresh Content

A requirement for any network caching system is the ability to ensure that users see the same content from a network cache as they would from the Web. Every Web page comprises several Web objects and each Web object has its own caching parameters, determined by content authors and HTTP standards (see the "HTTP Caching Standards" section). Thus, even a Web page with real-time objects typically has many other objects that are cacheable. Rotating ad banners and Common Gateway Interface (CGI)-generated responses are examples of objects that are typically noncacheable. Toolbars, navigation bars, GIFs, and JPEGs are examples of objects that are typically cacheable. Thus, for a given Web page, only a few dynamic objects need to be retrieved from the end server, while static objects can be fulfilled locally.

Cisco Content Engine products deliver fresh content by obeying the HTTP caching standards and by enabling administrators to have control over when content should be refreshed from origin Web servers.

HTTP Caching Standards

Cisco Content Engine products support HTTP 1.0 and 1.1, which specify caching parameters for each object on a Web page.

HTTP 1.0 allows content authors to enable a "Pragma: no cache" header field for any object that should not be cached and allows authors to enable content to be cached indefinitely.

HTTP 1.1 allows content authors to specify how long content is to be cached. For each object on a Web page, content authors can choose among the following caching attributes:

- Noncacheable
- OK to cache (the default setting)
- Explicit expiration date

HTTP 1.1 has a freshness revalidation mechanism called If-Modified-Since (IMS) to ensure that cached data is up to date. A content engine will send a lightweight IMS request to the end Web server when the content engine receives requests for cached content that has expired or IMS requests from clients where the cached content is older than a configured percentage of its maximum age. If the object has not been modified on the end server since the object was cached, the end server will return a lightweight message indicating that the content engine can deliver its cached copy to clients. If the object has been modified on the end server since the object was cached, the content engine will retrieve the fresh content. If the case of the client issuing an IMS request, and the content is less than a configured percentage of its maximum age, the content engine will serve the content without revalidating freshness.

Content Engine Content Freshness Controls

In addition to obeying HTTP caching standards, Cisco Content Engine products allow administrators to control the freshness of Web objects in a content engine. Content engines have a configurable parameter called the "freshness factor," which determines how fast or slow content expires. When an object is stored in cache, its time-to-live (TTL) value is calculated:

TTL value = (Current date – last modified date) * Configurable freshness factor

When an object "expires," based on its TTL value, the content engine will issue an IMS request the next time the object is requested (see "HTTP Caching Standards" section above for a description of the IMS freshness revalidation process).

If an administrator wants to adopt a conservative freshness policy, he or she can set the freshness factor to a small value (such as 0.05), so that objects expire more quickly. But the disadvantage to this approach is that IMS requests will be issued more frequently, consuming extra bandwidth. If an administrator wants to adopt a liberal freshness policy, the fresh factor can be set to a larger value, so that objects will expire more slowly and the IMS bandwidth overhead will be smaller.

Browser Freshness Controls

Finally, clients can always explicitly refresh content at any time by using the browser's reload/refresh button.

The "reload/refresh" command is a browser-triggered command to request a data refresh. A "reload/refresh" will issue a series of IMS requests asking for only data that has changed.

The "shift+reload/shift+refresh" command is an extension of the "reload/refresh" command. In correctly implemented browsers, this command always triggers a "pragma: no cache" rather than an IMS request. As a result, content engines are bypassed and all content is directly fulfilled by the end server.

Summary

By deploying a traffic localization solution, administrators can accelerate content delivery and optimize WAN bandwidth usage. Traffic localization is a traffic engineering problem because the network optimizes traffic flows based on specified parameters. Therefore, the first step in building a traffic localization solution is to ensure that your existing network supports traffic localization. This capability can be achieved by enabling transparent redirection technology, such as WCCP, in your network. Once the right network foundation is in place, content engines are added into strategic points within your existing network to complete the traffic localization solution.

Integrating content engines into your existing network infrastructure results in a network caching solution with a low cost of ownership, enabling you to cost-effectively deploy your caching services solution on a wide-scale basis and gain the benefits of caching services throughout your entire network.

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems Europe
11, Rue Camille Desmoulins
92782 Issy Les Moulineaux
Cedex 9
France
www.cisco.com
Tel: 33 1 58 04 60 00
Fax: 33 1 58 04 61 00

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems Australia, Pty., Ltd
Level 17, 99 Walker Street
North Sydney
NSW 2059 Australia
www.cisco.com
Tel: +61 2 8448 7100
Fax: +61 2 9957 4350

Cisco Systems has more than 190 offices in the following countries. Addresses, phone numbers, and fax numbers are listed on the Cisco.com Web site at www.cisco.com/go/offices.

Argentina • Australia • Austria • Belgium • Brazil • Canada • Chile • China • Colombia • Costa Rica • Croatia • Czech Republic • Denmark • Dubai, UAE
Finland • France • Germany • Greece • Hong Kong • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia
Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Singapore
Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela