Packet Flows in Common Cache Deployments

by Niall Doherty, Network Appliance, Inc.

April, 2004 | TR 3309

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Table of Contents

1. Introduction
2. Cache Deployments
2.1. Forward Proxy Configurations
2.2. Reverse Proxy Configurations
2.3. Summary
3. Transparent Redirection
3.1. Common Redirection Methods
3.2. Available Products
3.3. IP Spoofing
4. NetCache and Routing
4.1. Sending Packets
4.2. Fast IP
5. Packet Flow Examination
5.1. Forward Proxy: Browser Proxy Configuration
5.2. Forward Proxy: Transparent Redirection
5.3. Forward Proxy: Transparent Redirection and IP Spoofing
5.4. Reverse Proxy: Typical Server Load Balancing (SLB)
5.5. Reverse Proxy: SLB Using "Direct Server Return" (DSR)

1. Introduction

An application-level caching device—referred to herein as a "cache"—is an intermediary device located between a *client* and an *origin server*. It is deployed on the principle that multiple users request overlapping subsets of available content.

Here we are interested in those devices that understand application-level protocols (such as HTTP, RTSP, and FTP) and add value by *caching* (or *splitting*, in the case of live video streams) frequently accessed content. The purpose of this document is to examine the packet flows between the various devices involved in common deployments of these caches.

The material is presented in a generic manner while also providing some implementation-specific details for the NetCache® product from Network Appliance, Inc.

2. Cache Deployments

Caches can be deployed in either a forward proxy or a reverse proxy (aka acceleration) configuration—both of which utilize the same underlying caching technology.

In a forward proxy deployment, a "service provider" (e.g., traditional ISP or IT group in an enterprise environment) places a cache close to its end users with the goal of providing improvements in bandwidth utilization and response times. The cache retrieves content from many origin servers.

In a standard reverse proxy deployment, a content owner places a cache close to its origin servers with the goal of reducing the load upon them. The cache retrieves content only from these select origin servers.

In a "distributed" reverse proxy deployment—aka content delivery network (CDN)—multiple caches are placed at various locations close to different groups of end users. The goal here is to improve the response time of one's own content. Global server load balancing (GSLB) is a technology commonly used in these deployments: it directs end users to the closest "delivery point."

2.1. Forward Proxy Configurations

In forward proxy mode a client makes a request that is explicitly targeted at an origin server. The cache is deployed as a "middleware" device and handles the request on behalf of the client.

A key point about this method of deployment is that the cache is capable of determining the source of the content by examining information in the client's request.

2.1.1. Transparent Redirection

Transparent redirection refers to a configuration whereby a networking element located in the path of the client-server traffic flow intercepts all—or some portion of—that traffic and sends it to another device; in this case, a cache.

In this scenario, the client is unaware of the existence of the cache and believes it is sending packets directly to the server.

The cache can use the destination IP address of the client's packets or the application-level headers (e.g., Host header in HTTP) to determine the source of the content.

2.1.2. Browser Configuration

It is possible to configure all common Web and streaming clients (and many FTP clients) to use a proxy—either by directly entering the name of the proxy or by using a JavaScript automatic proxy configuration script.

In this scenario, the client communicates specifically with the proxy. It passes application-level information to the proxy—in this case, the cache—allowing the proxy to determine the source of the content.

2.2. Reverse Proxy Configurations

In reverse proxy mode the cache presents itself as the origin server—i.e., the client believes the cache is the source of the content.

A key point about this method of deployment is that the client request does not contain enough information for the cache to determine the actual origin server.

A reverse proxy configuration, therefore, must contain a set of rules that describe to the cache where to retrieve content from when it receives a request in reverse proxy mode.

2.3. Summary

Table 1. Summary of forward proxy and reverse proxy deployments.

	Forward Proxy	**Reverse Proxy**
Cache owned by...	Service provider	Content owner
Cache is located...	Close to end users	Close to origin servers (also close to end users, for CDN)
Improvements in...	Bandwidth utilization and response times	Load on origin servers(also response times, for CDN)
Source of content...	Many origin servers	Small number of select origin servers
Cache determination of origin server...	From client request	From configuration rules

Finally, it should be noted that it is possible to implement an acceleration deployment using a forward proxy mode configuration—for example, by placing a cache directly in front of an origin server in transparent redirection mode.

3. Transparent Redirection

Transparent redirection is concerned only with the interception of traffic originating from a client or server. Traffic originating from the cache (destined to either the client or the server) is entirely independent—i.e., when a cache sends a packet to a client or server, it should not be affected by the transparent redirection setup.

When a packet is transparently redirected to a cache, that cache receives a packet that is destined for its interface at the Ethernet/network layer but has a destination IP address of a different host. The TCP/IP network stack of the cache must be operating in *promiscuous mode* to accept these packets, since they would normally be rejected—i.e., the cache needs to accept and process packets for which it is does not appear to be the intended end destination (systems providing routing functionality do the same).

3.1. Common Redirection Methods

3.1.1. L2 Rewrite of Destination MAC Address

This is the most common and simplest method.

When a packet arrives at the redirection device, the destination IP address is that of the server, and the destination MAC address is that of the next hop on the path to the server.

To achieve redirection, the destination MAC address of the packet is changed to that of the cache, and the packet is then sent to the appropriate interface. The cache must be on the same network segment as the redirection device for this method to work.

3.1.2. Encapsulation Using IP-GRE

The second option is to use IP-GRE (encapsulation) to send the packet from the redirection device to the cache. This allows the cache and the redirection device to reside on different networks; in practice, however, this is rarely useful. While the IP-GRE method works, it may add extra processing overhead to the redirection device. If the redirection device supports both methods, then the MAC address rewriting method is recommended.

3.2. Available Products

3.2.1. Layer 4 Switches

A number of vendors (e.g., Foundry, Alteon) provide networking switches that support transparent redirection.

These switches are commonly known as "layer 4 switches" because "application-level" traffic is intercepted and redirected. Most of these devices are flexible enough to support redirection of inbound or outbound traffic, and they can usually make decisions on traffic distribution based on both the source and the destination IP addresses of the packets. Redirection is achieved using MAC address rewriting.

A disadvantage of this solution is that extra equipment needs to be installed on the network.

3.2.2. WCCP Routers/Switches

Cisco developed a protocol known as WCCP that allows caches to communicate with Cisco equipment and enables transparent redirection. It supports functionality similar to that of the layer 4 switches. There are two main differences:

- Application-level traffic configuration is done on the cache, and the cache communicates this to the *WCCP Router*. With a layer 4 switch, all configuration is performed on the switch itself.

- Since WCCP is supported on both Catalyst switches and a number of Cisco routers, there may be no need to purchase, install, and maintain new equipment.

While WCCP supports both the MAC address rewriting and IP-GRE redirection methods, particular routers/switches may implement only one specific method.

Also, note that enabling WCCP will incur some additional load on the router/switch, with IP-GRE generally being more burdensome than MAC address rewriting when both methods are available.

3.3. IP Spoofing

In the most basic form of transparent redirection, packets sent from a client to a server are intercepted and redirected to a cache.

The client is unaware of the presence of the cache and is expecting to receive packets from the server. To maintain an illusion of client-server end-to-end connectivity, the cache responds to the client using, as its source IP address, the destination IP address of the client's packet (i.e., the IP address of the server).

The cache is, in actuality, performing IP Spoofing on the *client-cache* connection.

The term *IP Spoofing*, while applying to the case above, is more commonly used to refer to IP Spoofing on the *cache-server* connection. In this case, the cache uses, as its source IP address, the source IP address of the client's packet (i.e., the IP address of the client) when communicating with the server.

The server perceives the packets as arriving from the *client's IP address*, and its responses will be sent directly to the client.

The implication here is that if IP Spoofing is desired for the cache-server connection, then it is not sufficient to redirect only the client requests: the server responses must also be redirected. Since HTTP responses typically contain greater amounts of data than HTTP requests, the additional load placed upon the redirection device should be considered.

IP Spoofing is trivial for a cache to support; it simply sets a different source IP address in packets it sends. Determining whether a particular network topology will be suitable for this configuration, however, may require detailed study and experimentation.

4. NetCache and Routing

4.1. Sending Packets

When a device sends a packet, it must

- Decide if the destination host is on a local network segment
- Choose an interface to send the packet through
- Discover the MAC address of either the destination host or the "next hop" (if the destination host is not on a local network segment)

Routing tables decide which interface to send the packet through. They also list any "default gateways" (routers) to use if the destination host is not on a local network segment. ARP is used to discover the MAC address of devices on a local network segment.

4.2. Fast IP

While devices typically cache the results of routing table lookups and ARP queries, it is desirable to reduce the need for these if possible.

The Fast IP option in NetCache (*enabled*, by default) attempts to optimize network routines by maintaining the following parameters for each open connection:

- Interface to send outgoing packets through
- MAC address of destination for outgoing packets

The parameters are extracted (and updated), on a per connection basis, from the most recent incoming packet that is received for that connection:

- The interface to be used for outgoing packets is set to the interface on which the most recent incoming packet arrived.
- The destination MAC address for outgoing packets is set to the source MAC address of the most recent incoming packet.

These parameters are set as follows:

- If another device initiates the connection, the SYN packet will be used to initialize the parameters.
- If NetCache initiates the connection (using normal ARP and routing table information), then the parameters are first set when the SYN-ACK packet arrives.

Some implications of having Fast IP enabled include the following:

- NetCache will only perform route lookups and ARP queries for the first packet that it sends on connections that it initiates.
- NetCache will always communicate with a client via the interface that the previous packet from that client arrived on. Packets are sent to the "last hop" of the incoming packet on the assumption that this device will know how to route the packet back to the client. This means routing descriptions for client networks are not required on NetCache.
- If asymmetric routing exists between a multihomed NetCache appliance and a server (i.e., the SYN-ACK packet follows a different route to the SYN packet), the first packet sent may use a different route than all subsequent packets.

5. Packet Flow Examination

5.1. Forward Proxy: Browser Proxy Configuration

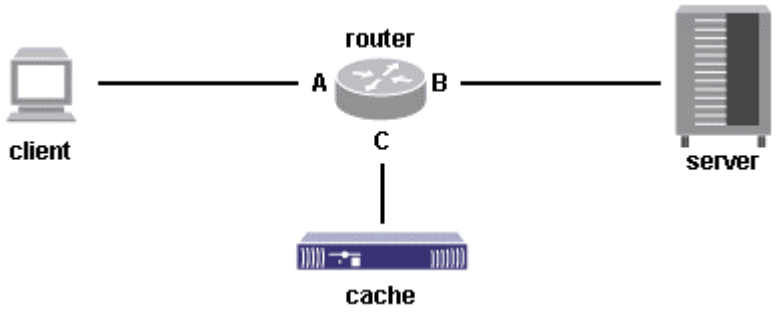

Figure 1. Forward proxy—direct client access to cache.

This first deployment considers a client communicating directly with a cache. The packet flows

demonstrate standard TCP/IP routing over Ethernet network segments.

Table 2. Client to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	cache-ip
Received by cache	router-C-MAC	cache-MAC	client-ip	cache-ip

Table 3. Cache to client.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-C-MAC	cache-ip	client-ip
Received by client	router-A-MAC	client-MAC	cache-ip	client-ip

Table 4. Cache to server.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-C-MAC	cache-ip	server-ip
Received by server	router-B-MAC	server-MAC	cache-ip	server-ip

Table 5. Server to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by server	server-MAC	router-B-MAC	server-ip	cache-ip
Received by cache	router-C-MAC	cache-MAC	server-ip	cache-ip

5.2. Forward Proxy: Transparent Redirection

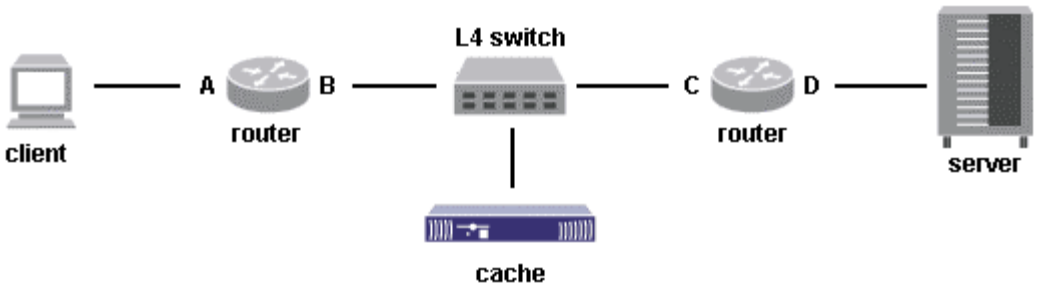

Figure 2. Transparent redirection.

This example shows how packets are modified in a transparent redirection scenario.

Table 6. Client to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	server-ip[1]
Received by cache	router-B-MAC	cache-MAC[2]	client-ip	server-ip

Note 1: The client sends packets directly to the server.

Note 2: The redirection device sets the destination MAC address to that of the cache.

Table 7. Cache to client.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-B-MAC	server-ip[3]	client-ip
Received by client	router-A-MAC	client-MAC	server-ip	client-ip

Note 3: When communicating with the client, the cache sets its source IP address to that of the server—i.e., it "spoofs" its source IP address for the cache-to-client packets.

Table 8. Cache to server.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-C-MAC	cache-ip	server-ip
Received by server	router-D-MAC	server-MAC	cache-ip	server-ip

Table 9. Server to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by server	server-MAC	router-D-MAC	server-ip	cache-ip
Received by cache	router-C-MAC	cache-MAC	server-ip	cache-ip

5.3. Forward Proxy: Transparent Redirection and IP Spoofing

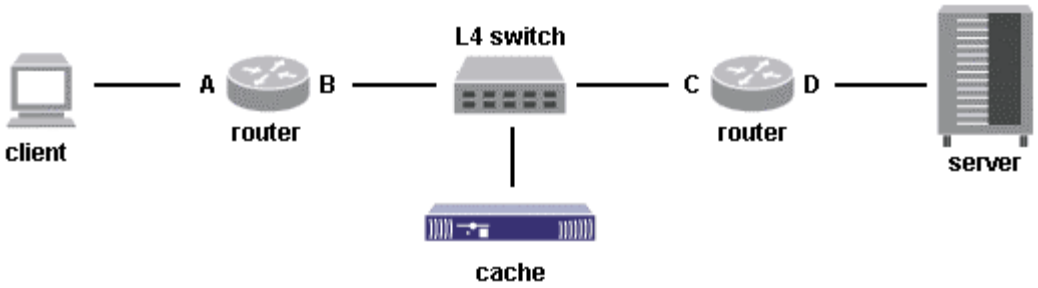

Figure 3. Transparent redirection and IP spoofing.

Here we introduce IP Spoofing on the cache-server connection.

Table 10. Client to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	server-ip
Received by cache	router-B-MAC	cache-MAC	client-ip	server-ip

Table 11. Cache to client.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-B-MAC	server-ip	client-ip
Received by client	router-A-MAC	client-MAC	server-ip	client-ip

Table 12. Cache to server.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-C-MAC	client-ip[4]	server-ip
Received by server	router-D-MAC	server-MAC	client-ip	server-ip

Note 4: When communicating with the server, the cache sets its source IP address to that of the client—i.e., it "spoofs" its source IP address for the cache-to-server packets.

Table 13. Server to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by server	server-MAC	router-D-MAC	server-ip	client-ip[5]
Received by cache	router-C-MAC	cache-MAC	server-ip	client-ip

Note 5: The server sends its responses directly to the client.

Interestingly, for this deployment, it is only possible to identify packets to and from the cache in a packet trace by examining MAC addresses; its IP address does not appear.

Finally, consider the consequences of enabling IP Spoofing on the cache-server connection without intercepting and redirecting returning traffic from the server:

- The cache would send a SYN packet to the server with the source IP address set to that of the client.
- The server would then reply, directly to the client, with a SYN-ACK packet.
- The client would receive this SYN-ACK packet and promptly ignore it, since it had no knowledge of ever attempting to initiate a connection with that server.

Clearly, the cache and server would not be capable of establishing a communication session. Redirection of returning traffic is, therefore, a necessity.

5.4. Reverse Proxy: Typical Server Load Balancing (SLB)

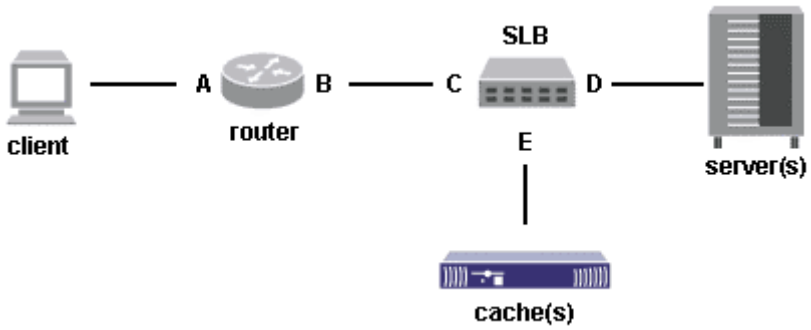

Figure 4. Typical server load balancing.

In a typical deployment of this kind, a single virtual IP (VIP) address on a server load balancer (SLB) device represents a number of hosts. A client sends packets to the VIP address, and the SLB device chooses which host receives the packets. The SLB device is responsible for ensuring packets sent from the cache to the client appear to come from the VIP address.

Table 14. Client to cache (A).

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	SLB-VIP
Received by cache	SLB-E-MAC[6]	cache-MAC	client-ip	cache-ip[7]

Note 6: Here, the source MAC address has been changed to that of the SLB device. Some SLB devices, when forwarding the packets from the client to the cache, may not change the source MAC address. This is shown in Table 15.

Note 7: The SLB device changes the destination IP address in the client's packets from the VIP address to the cache's IP address.

Table 15. Client to cache (B).

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	SLB-VIP
Received by cache	router-B-MAC[8]	cache-MAC	client-ip	cache-ip

Note 8: Here, the SLB device has not changed the source MAC address.

Table 16. Cache to client.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	SLB-E-MAC	cache-ip	client-ip
Received by client	router-A-MAC	client-MAC	SLB-VIP[9]	client-ip

Note 9: The SLB device changes the source IP address in the cache's packets from the cache's IP address to the VIP address.

Table 17. Cache to server.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	SLB-E-MAC	cache-ip	server-ip
Received by server	SLB-D-MAC	server-MAC	cache-ip	server-ip

Table 18. Server to cache.

	src MAC	dst MAC	src IP	dst IP
Sent by server	server-MAC	SLB-D-MAC	server-ip	cache-ip
Received by cache	SLB-E-MAC	cache-MAC	server-ip	cache-ip

5.5. Reverse Proxy: SLB Using "Direct Server Return" (DSR)

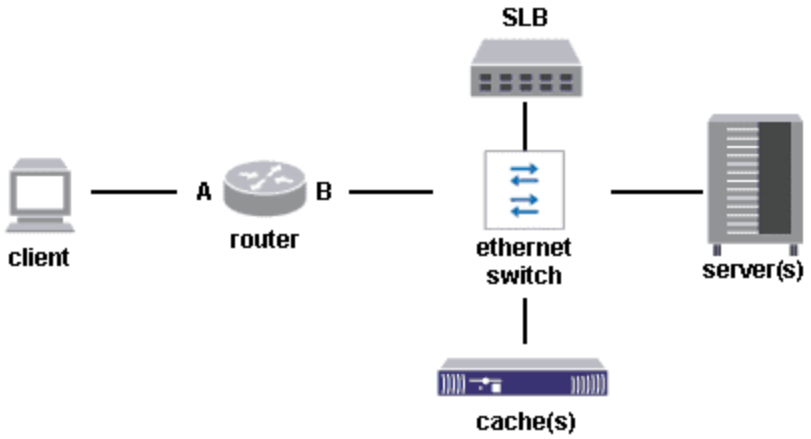

Figure 5. Reverse proxy mode using DSR.

The motivation behind this design is to reduce the load on the SLB device by only having packets from the client pass through it. Packets from the cache are sent directly to the client and do not pass through the SLB device. The cache, therefore, is responsible for ensuring its packets appear to come from the VIP address.

Table 19. Client to cache (A).

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	SLB-VIP
Received by SLB	router-B-MAC	SLB-MAC	client-ip	SLB-VIP
Sent by SLB	SLB-MAC[10]	cache-MAC	client-ip	SLB-VIP[11]
Received by cache	SLB-MAC	cache-MAC	client-ip	SLB-VIP

Note 10: Here, the source MAC address has been changed to that of the SLB device. Table 20 shows the situation if the SLB device does not change the source MAC address.

Note 11: In this scenario, the SLB device is configured to leave the destination IP address (the VIP address) unchanged when it sends the client's packets to the cache.

Table 20. Client to cache (B).

	src MAC	dst MAC	src IP	dst IP
Sent by client	client-MAC	router-A-MAC	client-ip	SLB-VIP
Received by SLB	router-B-MAC	SLB-MAC	client-ip	SLB-VIP
Sent by SLB	router-B-MAC[12]	cache-MAC	client-ip	SLB-VIP
Received by cache	router-B-MAC	cache-MAC	client-ip	SLB-VIP

Note 12: Here, the SLB device has not changed the source MAC address.

Table 21. Cache to client.

	src MAC	dst MAC	src IP	dst IP
Sent by cache	cache-MAC	router-B-MAC	SLB-VIP[13]	client-ip

Received by client	router-A-MAC	client-MAC	SLB-VIP	client-ip

Note 13: The cache replies directly to the client and must set its source IP address to that of the VIP address.

Table 22. Cache to server.

	src MAC	**dst MAC**	**src IP**	**dst IP**
Sent by cache	cache-MAC	server-MAC	cache-ip	server-ip
Received by server	cache-MAC	server-MAC	cache-ip	server-ip

Table 23. Server to cache.

	src MAC	**dst MAC**	**src IP**	**dst IP**
Sent by server	server-MAC	cache-MAC	server-ip	cache-ip
Received by cache	server-MAC	cache-MAC	server-ip	cache-ip

Table 24 describes how to configure the NetCache product appropriately for this type of server load balancing deployment.

Table 24. NetCache configuration for reverse proxy mode using DSR.

Configuration Option	**Value**
Transparency	Enabling transparency causes the network layer to accept packets that have a destination IP address of a different host. Responses will use this IP address as the source IP address.
HTTPS	If HTTPS acceleration is required, an inbound context rule must be matched.Note: If an inbound context rule is not matched, the connection is treated (incorrectly, in this case) as a transparent tunnel.
Acceleration	For an acceleration rule to match, the Incoming IP Address configuration option must match the destination IP address of the client's packets. In this type of deployment, the destination IP address will not be one of the configured interface addresses. A value of ** can be used; this causes a match for any destination IP address. It is more appropriate (and recommended), however, to specifically use the **VIP** address instead. Note: If an acceleration rule is not matched, the connection is treated (incorrectly, in this case) as a forward-proxy situation.
Fast IP	If the SLB device changes the source MAC address in the packets forwarded to NetCache, it is necessary to **disable** Fast IP so that packets from NetCache will not be sent back directly to the SLB device.
ACLs	If NetCache is deployed only for reverse proxy use, it is advisable to use an ACL of **deny not accel** so that NetCache cannot be used in forward proxy mode.