



Application Note

Multiple SIParator Distribution

26 May 2008

Table of Contents

1	MULTIPLE INGATE SIPARATOR SOLUTION	1
2	WHAT IS DNS SRV?	1
2.1	LOAD BALANCING WITH SRV	2
3	WHAT IS A SIP PROXY?	3
4	WHAT IS A B2BUA?	4
5	LOAD BALANCING SIPARATORS USING DNS SRV.....	5
6	LOAD BALANCING SIPARATORS USING A SIPARATOR AS A SIP PROXY	6
6.1	USING ONLY IP ADDRESSES	6
6.2	USING DNS A STANDARD	9

Tested versions: Ingate Firewall/SIParator/MEDIAtor version 4.6.2

Revision History:

Revision	Date	Author	Comments
	2008-05-26	Scott Beer	1 st draft

1 Multiple Ingate SIParator Solution

There are a number of applications for having multiple Ingate SIParators, spreading the SIP traffic between multiple SIParators, increasing size or scale of the deployment beyond the capacity of one single unit, providing a resilient and fault tolerant network and very large scale depolyments.

In each application, the solution is similar, to have multiple Ingate SIParators manage all of the traffic from the public Internet to a SIP server located on a secure LAN. There are two methods for providing this capability to the Ingate SIParator. First is by utilizing the service location capabilities within a DNS SRV record. Next is by having a Ingate SIParator act as a primary SIP Proxy server to distribute the SIP traffic over multiple SIParators.

2 What is DNS SRV?

The Domain Name System (DNS) associates various sorts of information with domain names; most importantly, it serves as the "phone book" for the Internet by translating human-readable computer hostnames, e.g. `www.ingate.com`, into the IP addresses, e.g. `88.131.69.225`, that networking equipment needs to deliver information. It also stores other information such as the list of mail exchange servers that accept email for a given domain. In providing a worldwide keyword-based redirection service, the Domain Name System is an essential component of contemporary Internet use.

Above all, DNS makes it possible to assign Internet names to organizations, independently of the physical routing hierarchy represented by the numerical IP address. Because of this, hyperlinks and Internet contact information can remain the same, whatever the current IP routing arrangements may be, and can take a human-readable form (such as "ingate.com"), which is easier to remember than the IP address `88.131.69.225`. People take advantage of this when they recite meaningful URLs and e-mail addresses without caring how the machine will actually locate them.

The Domain Name System distributes the responsibility for assigning domain names and mapping them to IP networks by allowing an authoritative server for each domain to keep track of its own changes, avoiding the need for a central registrar to be continually consulted and updated.

An SRV record or Service record is a category of data in the Domain Name System specifying information on available services. It is defined in RFC 2782. Newer internet protocols such as SIP and XMPP often require SRV support from clients.

An SRV record has the form:

`_Service._Proto.Name TTL Class SRV Priority Weight Port Target`

- Service:** the symbolic name of the desired service.
- Proto:** the protocol of the desired service; this is usually either TCP or UDP.
- Name:** the domain name for which this record is valid.
- TTL:** standard DNS time to live field.
- Class:** standard DNS class field (this is always IN).
- Priority:** the priority of the target host, lower value means more preferred.
- Weight:** A relative weight for records with the same priority.
- Port:** the TCP or UDP port on which the service is to be found.
- Target:** the canonical hostname of the machine providing the service.

An example SRV record might look like this using bind syntax:

```
_sip._udp.ingate.com. 86400 IN SRV 0 5 5060 sipserver.ingate.com.
```

This points to a server named sipserver.ingate.com listening on UDP port 5060 for SIP protocol connections. The priority given here is 0, and the weight is 5.

2.1 Load Balancing with SRV

The priority field is similar to an MX record's priority value. Clients always use the SRV record with the lowest-numbered priority value first, and only fall back to other records if the connection with this record's host fails. Thus a service may have a designated "fallback" server, which will only be used if the primary server fails. Only another SRV record, with a priority field value higher than the primary server's record, is needed.

If a service has multiple SRV records with the same priority value, clients use the weight field to determine which host to use. The weight value is relevant only in relation to other weight values for the service, and only among records with the same priority value.

In the following example, both the priority and weight fields are used to provide a combination of load balancing and backup service.

```
_sip._udp.ingate.com 86400 IN SRV 10 60 5060 bigsiparator.ingate.com
_sip._udp.ingate.com 86400 IN SRV 10 20 5060 smallsiparator1.ingate.com
_sip._udp.ingate.com 86400 IN SRV 10 10 5060 smallsiparator2.ingate.com
_sip._udp.ingate.com 86400 IN SRV 10 10 5070 smallsiparator2.ingate.com
_sip._udp.ingate.com 86400 IN SRV 20 0 5060 backupsiparator.ingate.com
```

The first four records share a priority of 10, so the weight field's value will be used by clients to determine which server (host and port combination) to contact. The sum of all four values is 100, so bigsiparator.ingate.com will be used 60% of the time. The two hosts smallsiparator1 and smallsiparator2 will be used for 20% of requests each, with half of the requests that are sent to smallsiparator2 (i.e. 10% of the total requests) going to port 5060 and the remaining half to port 5070. If bigsiparator is unavailable, these two remaining machines will share the load equally, since they will each be selected 50% of the time.

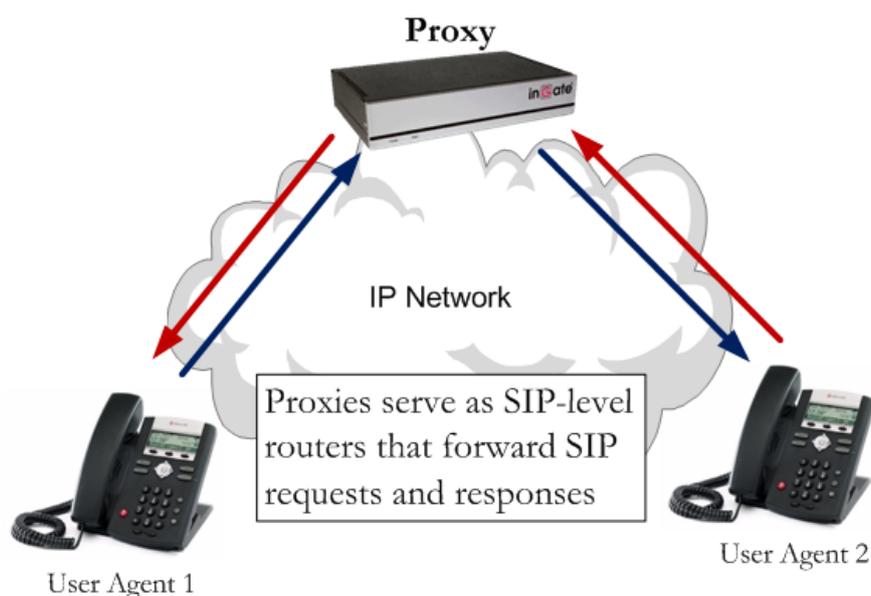
If all four servers with priority 10 are unavailable, the record with the next highest priority value will be chosen, which is backupbox.ingate.com. This might be a machine in another physical location, presumably not vulnerable to anything that would cause the first four hosts to become unavailable.

Several implementations relying on DNS also automatically update the DNS servers to optimize the load of servers or to remove servers due to service failures.

3 What is a SIP Proxy?

The SIP standard defines SIP proxies as “elements that route SIP requests to User Agent Servers (UAS) and SIP responses to User Agent Clients (UAC). A request may traverse several proxies on its way to a UAS. Each will make routing decisions, modifying the request before forwarding it to the next element. Responses will route through the same set of proxies traversed by the request in the reverse order.”

It is useful to view Proxy Servers as SIP-level routers that forward SIP requests and responses. However SIP proxies employ routing logic that is typically more sophisticated than just automatically forwarding messages based on a routing table. The SIP standard allows proxies to perform actions such as validate requests, authenticate users, fork requests, resolve addresses, cancel pending calls, Record-Route and Loose-Route, and detect and handle loops. The versatility of SIP proxies allows the system administrator to use the proxies for different purposes and in different locations in the network (such as edge proxy, core proxy and enterprise proxy). This versatility also allows for the creation of a variety of proxy policies, such as routing calls only for authenticated users that have no standing debt to the network service provider operating the proxy. Proxies can be placed at the network of the service provider or at the enterprise premises.



4 What is a B2BUA?

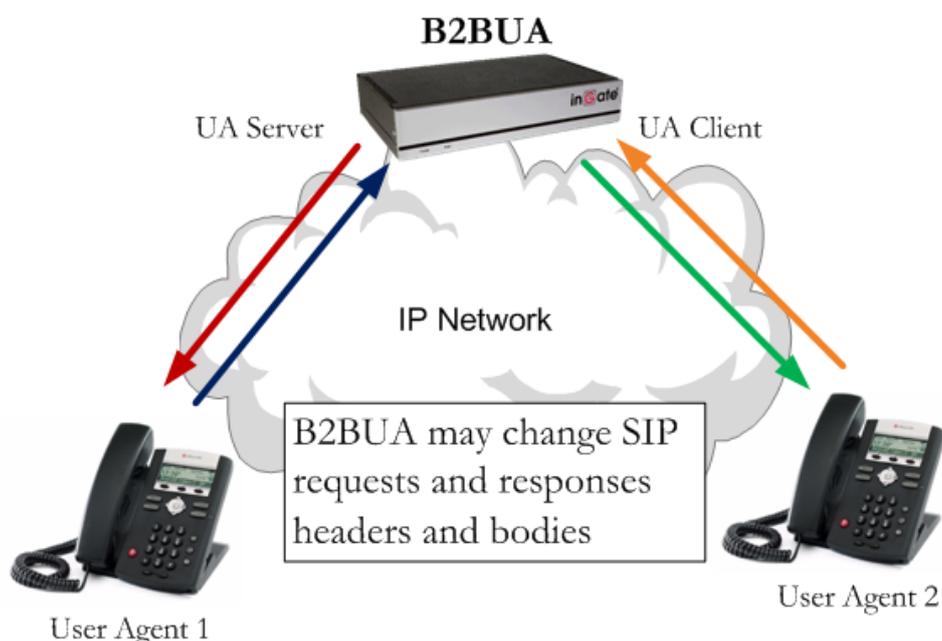
The Back-to-Back User Agent (B2BUA) acts as a user agent to both ends of a SIP call. The B2BUA is responsible for handling all SIP signalling between both ends of the call, from call establishment to termination. Each call is tracked from beginning to end, allowing the operators of the B2BUA to offer value-added features to the call. A B2BUA takes what is traditionally a SIP end-to-end call and mediates it through a central SIP server. The B2BUA enables service providers and enterprises to manage and track a call from beginning to end, and integrate and offer new value added features.

To SIP clients, the B2BUA acts as a User Agent server on one side and as a User Agent client on the other (back-to-back) side. The basic implementation of a B2BUA is defined in RFC 3261. The B2BUA may provide the following functionalities:

- Centralized call management
- Interworking with alternative networks
- SIP--based VoIP interworking between LAN and WAN
- Management and monitoring of the entire call state
- Cloaking of endpoint location
- Centralized call management

Because it maintains call state for all SIP calls it handles, failure of a B2BUA affects all these calls. Often, B2BUAs also terminate and bridge the media streams to have full control over the whole session.

It is important to note that the B2BUA within the Ingate SIParator can also be used to mediate the SIP end-to-end call through a central SIP server. Mediating the Service Provider call and also distributing the call flow to the other SIParators for Load Balancing.



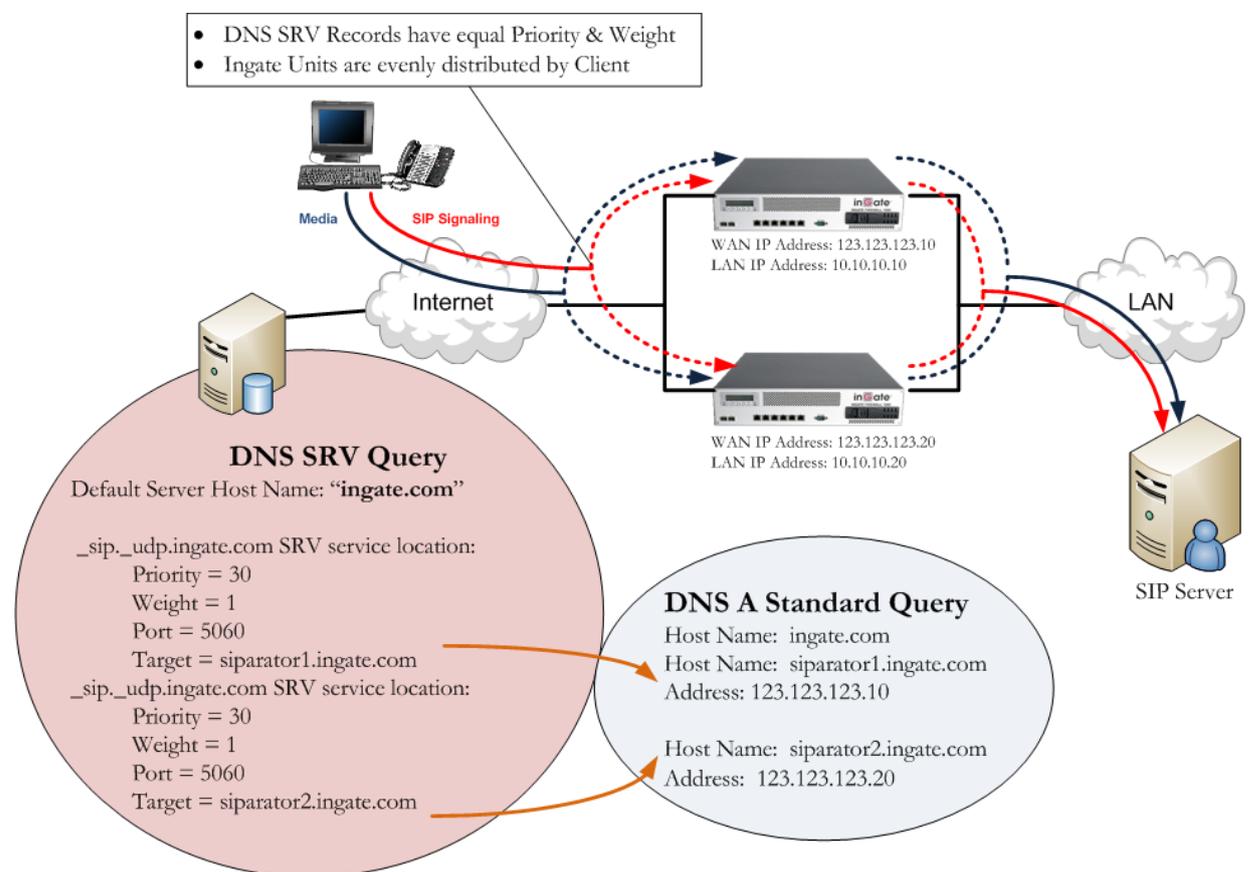
5 Load Balancing SIParators Using DNS SRV

In the following example, both the priority and weight fields are used to provide load balancing for SIP services on the ingate.com domain.

```
_sip._udp.ingate.com 86400 IN SRV 30 1 5060 smallsiparator1.ingate.com
_sip._udp.ingate.com 86400 IN SRV 30 1 5060 smallsiparator2.ingate.com
```

The two records share a priority of 30, and the weight field's value are also equal, these will be used by clients to determine which server (host and port combination) to contact. The sum of all four values is 60, so siparator1.ingate.com will be used 50% of the time, while siparator2.ingate.com will be used 50% of the time as the weight field's value are equal. If siparator1 is unavailable, the remaining machine will take all the load.

DNS SRV Load Balancing Example



Other considerations should be noted, as this document does not describe Network distribution, resiliency and failover scenarios in combination with having multiple Ingates. If this solution is not properly distributed over a network then it is possible that the Ingate SIParator may have service degradation. Providing DNS Server updates with current network information will assist in providing a resilient environment.

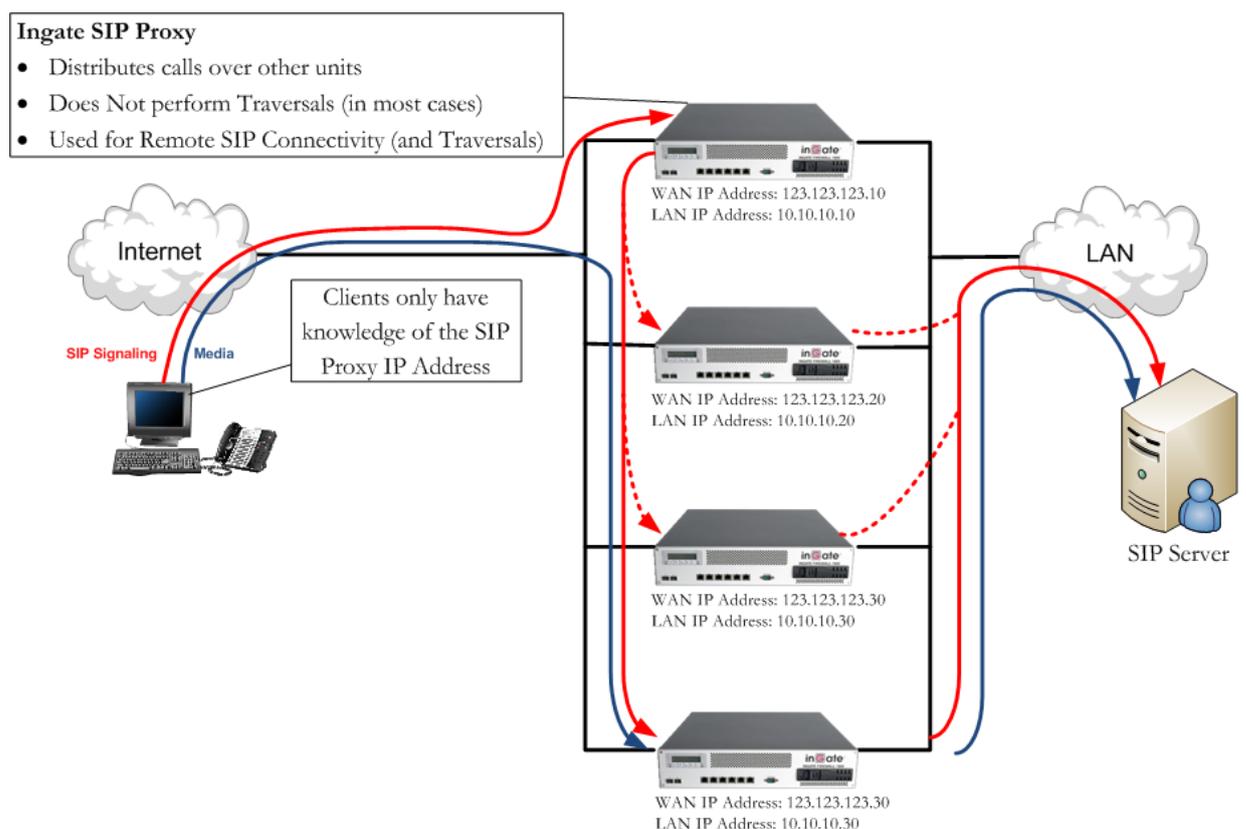
6 Load Balancing SIParators Using a SIParator as a SIP Proxy

In this example there is an Ingate SIParator acting as a Proxy Server. This Proxy Server is being used as SIP-level router that is forwarding SIP requests and responses to distribute the call flow over several other SIParators. The SIParator SIP Proxy is employing routing logic that is typically more sophisticated than just automatically forwarding messages based on a routing table. The SIParator is can perform the same Load Balancing function in either direction.

6.1 Using Only IP Addresses

SIParator SIP Proxy Example

Using only IP addresses, the SIP Client on the internet directs all of its traffic to the Ingate SIP Proxy. The Ingate SIP Proxy, using either the Proxy capabilities or the B2BUA capabilities, distribute the calls among the other Ingate SIParators. These SIParators perform the traversal to the private network where the SIP Server resides.



Other considerations should be noted, as this document does not describe Network distribution, resiliency and failover scenarios in combination with having multiple Ingates. If this solution is not properly distributed over a network then it is possible that the Ingate SIP Proxy is a single point of failure in this solution.

SIParator SIP Proxy using DNS Override for SIP Requests Example

This example shows how the Ingate SIP Proxy can distribute calls among the other SIParators. Using "DNS Override for SIP Requests", the Ingate matches the Domain IP address of the incoming SIP signaling then directing the traffic to the other SIParator with similar capabilities and behavior as DNS SRV, where the Ingate can assign priority and weight. Both the priority and weight fields are used to provide a combination of load balancing and backup service.

Even Load Distribution

This "DNS Override For SIP Requests" example shows how to distribute call evenly over the other SIParators.

Domain	Relay To						Delete
	DNS Name or IP Address	IP Address	Port	Transport	Priority	Weight	
+ 10.10.10.10	10.10.10.20	10.10.10.20		UDP	30	1	<input type="checkbox"/>
	10.10.10.30	10.10.10.30		UDP	30	1	<input type="checkbox"/>
	10.10.10.40	10.10.10.40		UDP	30	1	<input type="checkbox"/>
+ 123.123.123.10	123.123.123.20	123.123.123.20		UDP	30	1	<input type="checkbox"/>
	123.123.123.30	123.123.123.30		UDP	30	1	<input type="checkbox"/>
	123.123.123.40	123.123.123.40		UDP	30	1	<input type="checkbox"/>

The first three records share a priority of 30, so the weight field's value will be used by clients to determine which server (host and port combination) to contact. The sum of all three values is 3, each having a value of 1, so they are all equal, thus they will share the load equally, since they will each be selected 1/3 of the time.

Similar characteristics can be applied in the other direction, directing traffic to the Ingate SIP Proxy 10.10.10.10 and having the Ingate distribute calls to the other SIParators.

Dynamic Load Distribution

This "DNS Override For SIP Requests" example shows how to distribute call dynamically over the other SIParators.

The screenshot shows a web interface for configuring SIP traffic. The navigation menu includes: Administration, Basic Configuration, Network, SIP Services, SIP Traffic (highlighted), Failover, Virtual Private Networks, Quality of Service, Logging and Tools, and About. Below this, a sub-menu includes: SIP Methods, Filtering, User Database, Authentication and Accounting, Dial Plan, Routing (highlighted), Time Classes, and SIP Status. The main content area is titled "DNS Override For SIP Requests (Help)".

Domain	Relay To						Delete
	DNS Name or IP Address	IP Address	Port	Transport	Priority	Weight	
+ 10.10.10.10	10.10.10.20	10.10.10.20		UDP	30	60	<input type="checkbox"/>
	10.10.10.30	10.10.10.30		UDP	30	40	<input type="checkbox"/>
	10.10.10.40	10.10.10.40		UDP	40	1	<input type="checkbox"/>
+ 123.123.123.10	123.123.123.20	123.123.123.20		UDP	30	60	<input type="checkbox"/>
	123.123.123.30	123.123.123.30		UDP	30	40	<input type="checkbox"/>
	123.123.123.40	123.123.123.40		UDP	40	1	<input type="checkbox"/>

The first two records share a priority of 30, so the weight field's value will be used by clients to determine which server (host and port combination) to contact. The sum of all four values is 100, so 123.123.123.20 will be used 60% of the time. The 123.123.123.30 will be used for 40% of requests each.

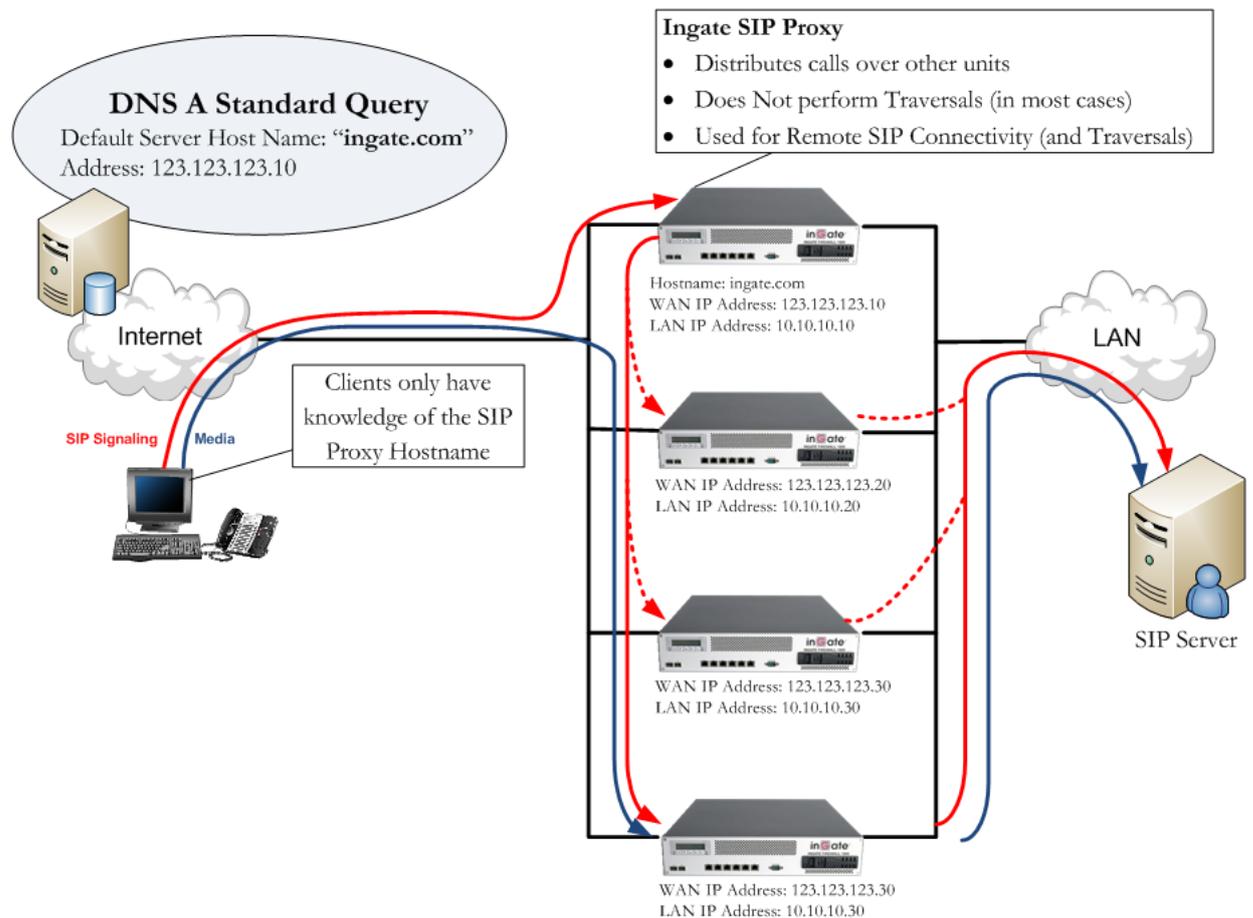
If both servers with priority 30 are unavailable, the record with the next highest priority value will be chosen, which is 123.123.123.40. This might be a machine in another physical location, presumably not vulnerable to anything that would cause the first two hosts to become unavailable.

Similar characteristics can be applied in the other direction, directing traffic to the Ingate SIP Proxy 10.10.10.10 and having the Ingate distribute calls to the other SIParators.

6.2 Using DNS A Standard

SIParator SIP Proxy Example

Using a Hostname or Domain, the SIP Client on the internet performs a DNS A Standard Lookup to resolve the Domain IP address. This IP address resolved from the DNS A Lookup resolves to the Ingate SIP Proxy. The SIP Client then directs all of its traffic to the Ingate SIP Proxy. The Ingate SIP Proxy, using either the Proxy capabilities or the B2BUA capabilities, distribute the calls among the other Ingate SIPParators. These SIPParators perform the traversal to the private network where the SIP Server resides.



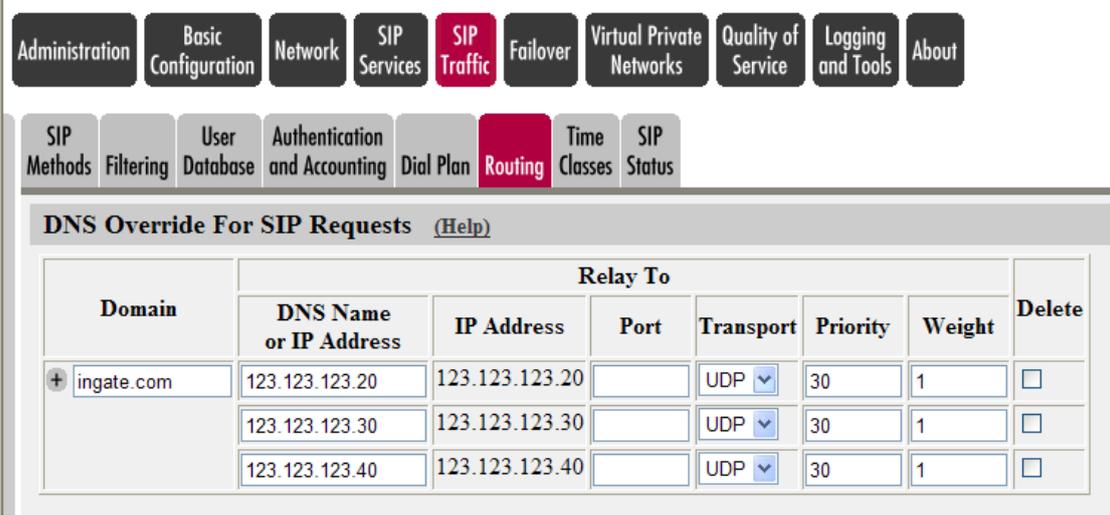
Other considerations should be noted, as this document does not describe Network distribution, resiliency and failover scenarios in combination with having multiple Ingates. If this solution is not properly distributed over a network then it is possible that the Ingate SIP Proxy is a single point of failure in this solution.

SIParator SIP Proxy using DNS Override for SIP Requests Example

This example shows how the Ingate SIP Proxy can distribute calls among the other SIParators. Using "DNS Override for SIP Requests", the Ingate matches the Domain IP address of the incoming SIP signaling then directing the traffic to the other SIParator with similar capabilities and behavior as DNS SRV, where the Ingate can assign priority and weight. Both the priority and weight fields are used to provide a combination of load balancing and backup service.

Even Load Distribution

This "DNS Override For SIP Requests" example shows how to distribute calls evenly over the other SIParators.



The screenshot shows the Ingate SIP Proxy configuration interface. The top navigation bar includes tabs for Administration, Basic Configuration, Network, SIP Services, SIP Traffic (highlighted), Failover, Virtual Private Networks, Quality of Service, Logging and Tools, and About. Below this, a sub-menu includes SIP Methods, Filtering, User Database, Authentication and Accounting, Dial Plan, Routing (highlighted), Time Classes, and SIP Status. The main content area is titled "DNS Override For SIP Requests (Help)" and contains a table with the following structure:

Domain	Relay To						Delete
	DNS Name or IP Address	IP Address	Port	Transport	Priority	Weight	
+ ingate.com	123.123.123.20	123.123.123.20		UDP	30	1	<input type="checkbox"/>
	123.123.123.30	123.123.123.30		UDP	30	1	<input type="checkbox"/>
	123.123.123.40	123.123.123.40		UDP	30	1	<input type="checkbox"/>

The first three records share a priority of 30, so the weight field's value will be used by clients to determine which server (host and port combination) to contact. The sum of all three values is 3, each having a value of 1, so they are all equal, thus they will share the load equally, since they will each be selected 1/3 of the time.

Since, SIP Requests from the SIP Server will not have its own domain, as it would be calling itself, the outgoing domain can also be applied here to distribute calls over the other SIParators.

Dynamic Load Distribution

This "DNS Override For SIP Requests" example shows how to distribute calls dynamically over the other SIParators.

The screenshot shows the Asterisk configuration interface. The top navigation bar includes: Administration, Basic Configuration, Network, SIP Services, SIP Traffic (selected), Failover, Virtual Private Networks, Quality of Service, Logging and Tools, and About. Below this, a sub-menu shows: SIP Methods, Filtering, User Database, Authentication and Accounting, Dial Plan, Routing (selected), Time Classes, and SIP Status. The main content area is titled "DNS Override For SIP Requests (Help)". It contains a table with the following structure:

Domain	Relay To						Delete
	DNS Name or IP Address	IP Address	Port	Transport	Priority	Weight	
+ ingate.com	123.123.123.20	123.123.123.20		UDP	30	60	<input type="checkbox"/>
	123.123.123.30	123.123.123.30		UDP	30	30	<input type="checkbox"/>
	123.123.123.40	123.123.123.40		UDP	30	10	<input type="checkbox"/>

The first two records share a priority of 30, so the weight field's value will be used by clients to determine which server (host and port combination) to contact. The sum of all four values is 100, so 123.123.123.20 will be used 60% of the time. The 123.123.123.30 will be used for 40% of requests each.

If both servers with priority 30 are unavailable, the record with the next highest priority value will be chosen, which is 123.123.123.40. This might be a machine in another physical location, presumably not vulnerable to anything that would cause the first two hosts to become unavailable.

Since, SIP Requests from the SIP Server will not have it's own domain, as it would be calling itself, the outgoing domain can also be applied here to distribute calls over the other SIParators.