**DNS Configuration Types**

Most DNS servers are schizophrenic - they may be masters (authoritative) for some [zones](http://www.zytrax.com/books/dns/apa/zones.html), slaves for others and provide caching or forwarding for all others. Many observers object to the concept of DNS **types** partly because of the schizophrenic behaviour of most DNS servers and partly to avoid confusion with the name.conf zone parameter 'type' which only allows master, slave, stub, forward, hint). Nevertheless, the following terms are commonly used to describe the primary function or requirement of DNS servers.

**Notes:**

1. Running any DNS server that supports recursive queries received from any or all users (an Open DNS) is an Extremely Bad Idea ™. While an Open DNS may look like a friendly and neighbourly thing to do such a server may be used in DDoS attacks and carries a significantly increased risk of cache poisoning. It is always possible to define the range of IP addresses that are allowed or permitted to use the recursive feature of any DNS server. Such a server is termed Closed. The various configurations have been modified to ensure that the DNS stays Closed to non-permitted users.
2. One of the basic rules of security is that only the minimum services necessary to meet the objectives should be deployed. This means that a secure DNS server should provide only a single function, for instance, authoritative only, or caching only, not both capabilities in the same server. This is a correct but idealistic position, generally possible only in larger organizations. In practice many of us run mixed mode DNS servers. While much can be done to mitigate any security implications it must always be accepted that, in mixed configurations, increased risk is the downside of flexibility.

**Contents**

1. [4.1 Master (a.k.a. Primary) DNS Server](http://www.zytrax.com/books/dns/ch4/#master)
2. [4.2 Slave (a.k.a Secondary) DNS Server](http://www.zytrax.com/books/dns/ch4/#slave)
3. [4.2.1 But Slaves can also be Masters](http://www.zytrax.com/books/dns/ch4/#slave-master)
4. [4.3 Caching (a.k.a. hint) DNS Server](http://www.zytrax.com/books/dns/ch4/#caching)
5. [4.4 Forwarding (a.k.a Proxy, Client, Remote) DNS Server](http://www.zytrax.com/books/dns/ch4/#forwarding)
6. [4.5 Stealth (a.k.a. DMZ or Hidden Master) DNS Server](http://www.zytrax.com/books/dns/ch4/#stealth)
7. [4.6 Authoritative Only DNS Server](http://www.zytrax.com/books/dns/ch4/#authoritative)
8. [4.7 Split Horizon DNS Server](http://www.zytrax.com/books/dns/ch4/#split)

**4.1 Master (Primary) Name Servers**

A Master DNS defines one or more [zone files](http://www.zytrax.com/books/dns/apa/zones.html) for which this DNS is **Authoritative** ('type master'). The zone has been delegated (via an [NS Resource Record](http://www.zytrax.com/books/dns/ch8/ns.html)) to this DNS.

The term **master** was introduced with BIND 8.x and replaced the term 'primary'.

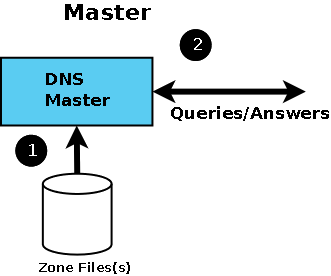


Diagram 1 DNS Master

Master status is defined in BIND by including 'type master' in the zone declaration section of the [named.conf file](http://www.zytrax.com/books/dns/apa/conf.html) as shown by the following fragment.

// example.com fragment from named.conf

// defines this server as a zone master

zone "example.com" in{

type master;

file "pri.example.com";

};

**Notes:**

1. The terms Primary and Secondary DNS entries in Windows TCP/IP network properties mean nothing, they may reflect the 'master' and 'slave' name-server or they may not - you decide this based on operational need, not BIND configuration.
2. It is important to understand that a zone 'master' is simply a server which gets its zone data from a local source as opposed to a 'slave' which gets its zone data from an external (networked) source (typically the 'master' but not always). This apparently trivial point means that you can have any number of 'master' servers for any zone if it makes operational sense. You have to ensure (by a manual or other process) that the zone files are synchronised but apart from this there is nothing to prevent it.
3. Just to confuse things still further you may run across the term 'Primary Master' this has a special meaning in the context of [dynamic DNS updates](http://www.zytrax.com/books/dns/ch2/index.html#dyn-update) and is defined to be the name server that appears in the [SOA RR record](http://www.zytrax.com/books/dns/ch8/soa.html).

When a master DNS receives [Queries](http://www.zytrax.com/books/dns/apa/query.html) for a zone for which it is authoritative then it will respond as 'Authoritative' (AA bit is set in a query response).

If a DNS server receives a query for a zone for which it is neither a Master nor a [Slave](http://www.zytrax.com/books/dns/ch4/#slave) then it will act as configured (in BIND this behaviour is defined in the [named.conf file)](http://www.zytrax.com/books/dns/apa/conf.html):

1. If [caching behaviour](http://www.zytrax.com/books/dns/ch4/#cache) is permitted and recursive queries are allowed the server will completely answer the request or return an error.
2. If [caching behaviour](http://www.zytrax.com/books/dns/ch4/#cache) is permitted and Iterative (non-recursive) queries are allowed the server can respond with the complete answer (if it is already in the cache because of another request), a referral or return an error.
3. If caching behaviour is NOT permitted (an ['Authoritative Only'](http://www.zytrax.com/books/dns/ch4/#authoritative) DNS server) the server will return a referral or an error.

A master DNS server can NOTIFY zone changes to defined (typically slave) servers - this is the default behaviour. NOTIFY messages ensure zone changes are rapidly propagated to the slaves (interrupt driven) rather than rely on the slave server periodically polling for changes. The BIND default is to notify the servers defined in [NS records](http://www.zytrax.com/books/dns/ch8/ns.html) for the zone - except itself, obviously.

A zone master can be 'hidden' (only one or more of the slaves know of its existence). There is no requirement in such a configuration for the master server to appear in an NS RR for the domain. The only requirement is that two (or more) name servers support the zone. Both servers could be any combination of master-slave, slave-slave or even master-master.

If you are running [Stealth Servers](http://www.zytrax.com/books/dns/ch4/#stealth) and wish them to be notified you will have to add an [also-notify parameter](http://www.zytrax.com/books/dns/ch7/xfer.html#also-notify) as shown in the BIND [named.conf](http://www.zytrax.com/books/dns/apa/conf.html) file fragment below:

// example.com fragment from named.conf

// defines this server as a zone master

// 192.168.0.2 is a stealth server NOT listed in a NS record

zone "example.com" in{

type master;

also-notify {192.168.0.2;};

file "pri/pri.example.com";

};

You can turn off all NOTIFY operations by specifying ['notify no'](http://www.zytrax.com/books/dns/ch7/xfer.html#notify) in the zone declaration.

Example configuration files for a master DNS [are provided](http://www.zytrax.com/books/dns/ch6/index.html#master).

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.2 Slave Name Servers**

A Slave DNS gets its zone data using a zone transfer operation (typically from a zone master) and it will respond as authoritative for those zones for which it is defined to be a 'slave' and for which it has a currently valid zone configuration. It is impossible to determine from a query result that it came from a zone master or slave.

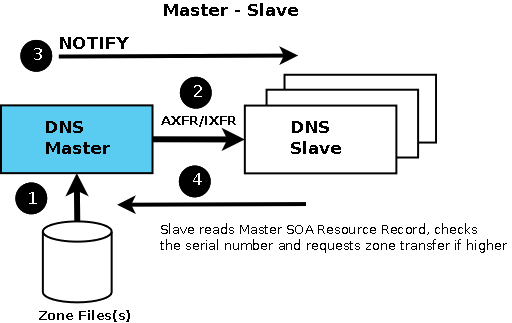


Diagram 2 DNS Slave Server

The term 'slave' was introduced with BIND 8.x and replaced the term 'secondary'.

There can be any number of slave DNS's for any given zone.

Slave status is defined in BIND by including 'type slave' in the zone declaration section of the [named.conf file](http://www.zytrax.com/books/dns/apa/conf.html) as shown by the following fragment.

// example.com fragment from named.conf

// defines this server as a zone slave

zone "example.com" in{

type slave;

file "sec/sec.example.com";

masters {192.168.23.17;};

};

**Notes:**

1. The master DNS for each zone is defined in the 'masters' statement of the zone clause and allows slaves to refresh their zone record when the 'expiry' parameter of the [SOA Record](http://www.zytrax.com/books/dns/ch8/soa.html) is reached. If a slave cannot reach the master DNS when the 'expiry' time has been reached it will stop responding to requests for the zone. It will not use time-expired data.
2. The file parameter is optional and allows the slave to write the transferred zone to disc and hence if BIND is restarted before the 'expiry' time the Slave server will use the saved data. In large DNS systems this can save a considerable amount of network traffic.

Assuming NOTIFY is allowed in the master DNS for the zone (the default behaviour) then zone changes are propagated to all the servers defined with [NS Records](http://www.zytrax.com/books/dns/ch8/ns.html) in the zone file. Other acceptable NOTIFY sources can be defined using the [also-notify](http://www.zytrax.com/books/dns/ch7/xfer.html#also-notify) parameter in named.conf.

Example configuration files for a slave DNS [are provided](http://www.zytrax.com/books/dns/ch6/index.html#slave).

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.2.1 But Slaves can also be Masters**

Oh, stop this pain. This section can get a bit confusing. Read it only when accompanied by your favorite keep-me-awake-cos-I-can't-take-anymore-of-this-stuff beverage.

The definition of a slave server is simply that it gets its zone data via zone transfer, whereas a master gets its zone data from a local file system. The source of the zone transfer could just as easily be another slave as a master.

So what sane human would want to do that?

1. Assume you want to hide your master servers in, say, a [stealth](http://www.zytrax.com/books/dns/ch4/#stealth) configuration then at least one slave server will sit on the public side of a firewall, or similar configuration, providing perimeter defence. To provide resilience you would need two or more such public slaves. The second slave can be updated from the same master as the first or it could be updated from the slave server - we'll call it the 'boss' slave to avoid getting into tortuous terminology (is it a master-slave or a slave-master?). To configure this miracle the second slave server would define the 'boss' slave's IP in its [masters](http://www.zytrax.com/books/dns/ch7/zone.html#masters) statement. When the 'boss' slave has sucessfully transfered a zone file (from the master) it will send out NOTIFY messages (the default) unless configured not to do so. This type of configuration will marginally increase latency for updating the zone on the second slave - but that may be more than offset by increased stealth.
2. In a DNSSEC environment the master will likely have all kinds of whizzo dodads concerned with keeping keys secure. Whereas DNSSEC slaves simply send the data in the zone file in response to queries and have no requirements for secure key maintenance. Hidden master configurations will become increasingly the norm in this environment.

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.3 Caching Name Servers (Resolver)**

A DNS Caching Server (frequently called a Resolver) obtains information from another server (a Zone Master) in response to a host query and then saves (caches) the data locally. On a second or subsequent request for the same data the Caching Server (Resolver) will respond with its locally stored data (the cache) until the [time-to-live (TTL)](http://www.zytrax.com/books/dns/apa/ttl.html) value of the response expires, at which time the server will refresh the data from the zone master.

If the caching server (resolver) obtains its data directly from a zone master it will respond as 'authoritative', if the data is supplied from its cache the response is 'non-authoritative'.

The default BIND behaviour is to cache and this is associated with the [recursion](http://www.zytrax.com/books/dns/ch7/queries.html#recursion) parameter (the default is 'recursion yes'). There are many configuration examples which show caching behaviour being defined using a *type hint* statement in a zone declaration. These configurations confuse two distinct but related functions. If a server is going to provide caching services then it must support [recursive queries](http://www.zytrax.com/books/dns/ch2/index.html#recursive) and recursive queries need access to the root servers which is provided via the 'type hint' statement.

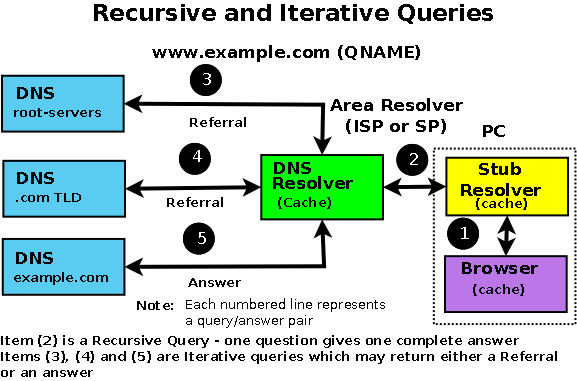


Diagram 3 DNS Resolver (Recursive Server)

A caching server will typically have a [named.conf file](http://www.zytrax.com/books/dns/apa/conf.html) which includes the following fragment:

// options section fragment of named.conf

// recursion yes is the default and may be omitted

options {

directory "/var/named";

version "not currently available";

recursion yes;

};

// zone section

....

// the DOT indicates the root domain = all domains

zone "." IN {

type hint;

file "root.servers";

};

**Notes:**

1. BIND defaults to [recursive queries](http://www.zytrax.com/books/dns/ch2/index.html#recursive) which by definition provides caching behaviour. The named.conf [recursion](http://www.zytrax.com/books/dns/ch7/queries.html#recursion) parameter controls this behaviour.
2. The zone '.' is shorthand for the root domain which translates to 'any domain not defined as either a master or slave in this named.conf file'.
3. cache data is discarded when BIND is restarted.

The most common DNS server caching configurations are:

* A DNS server acting as master or slave for one or more zones (domains) and as cache server for all other requests. A general purpose DNS server.
* A caching only local server - typically used to minimise external access or to compensate for slow external links. This is sometimes called a Proxy server though we prefer to associate the term with a [Forwarding server](http://www.zytrax.com/books/dns/ch4/#forwarding)

To cache or not is a crucial question in the world of DNS. BIND is regarded as the reference implementation of the DNS specification. As such it provides excellent - if complex to configure - functionality. The down side of generality is suboptimal performance on any single function - in particular caching involves a non-trivial performance overhead.

For general usage the breadth of BIND functionality typically offsets any performance concerns. However if the DNS is being 'hit' thousands of times per second performance is a major factor. There are now [a number of alternate Open Source DNS servers](http://www.zytrax.com/books/dns/apc/) some of which stress performance. These servers typically do NOT provide caching services (they are said to be ['Authoritative only' servers)](http://www.zytrax.com/books/dns/ch4/#authoritative).

Example configuration files for a caching DNS [are provided](http://www.zytrax.com/books/dns/ch6/index.html#caching).

**Note:** The response to a query is Authoritative under three conditions:

1. The response is received from a Zone master.
2. The response is received from a Zone slave with non time-expired zone data.
3. The response is received by a caching server directly from either a Zone master or slave. If the response is supplied from the cache it is not authoritative.

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.4 Forwarding (a.k.a Proxy) Name Servers**

A forwarding (a.k.a. Proxy, Client, Remote) server is one which simply forwards requests to another DNS and caches the results. On its face this looks like a pretty pointless exercise. It is, however, a frequently undervalued and extremely useful configuration in a number of situations:

1. Where access to the external network is slow or expensive:
   1. Local DNS caching - results are cached in the forwarding server so that frequently requested domains will provide fast results from the cache.
   2. The Remote (forwarded to) DNS server provides recursive query support resulting in a single query across the network (from the forwarding DNS to the 'forwared to' DNS) thus reducing traffic congestion (on busy networks), traffic volume (on expensive networks) and increasing performance (on slow networks).
2. Forwarding servers also can be used to ease the burden of local administration by providing a single point at which changes to remote name servers may be managed, rather than having to update all hosts. Thus, all hosts in a particular network section or area can be configured to point to a fixed forwarding DNS which can be configured to stream DNS traffic as desired and changed over time with minimal effort.
3. Sanitizing traffic. Especially in larger private networks it may be sensible to stream DNS traffic for local domain access by forwarding to the local DNS servers while forwarding external DNS requests to a **dirty** or hardened caching DNS (or resolver).
4. Forwarding can also be used as part of a [Split Server](http://www.zytrax.com/books/dns/ch4/#stealth) configuration for perimeter defence.

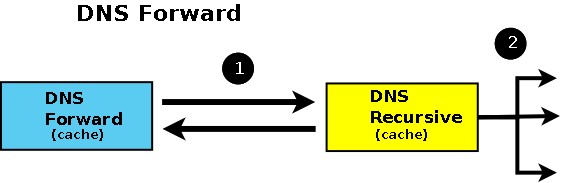


Diagram 4 - DNS Forvarding Server

BIND allows configuration of forwarding using the [forward](http://www.zytrax.com/books/dns/ch7/queries.html#forward) and [forwarders](http://www.zytrax.com/books/dns/ch7/queries.html#forwarders) parameters either at a 'global' level (in an options section) or on a per-zone basis in a zone section of the [named.conf file](http://www.zytrax.com/books/dns/apa/conf.html). Both configurations are shown in the examples below:

**Global Forwarding - All Requests**

// options section fragment of named.conf

// forwarders can have multiple choices

options {

directory "/var/named";

version "not currently available";

forwarders {10.0.0.1; 10.0.0.2;};

forward only;

};

// zone file sections

....

**Per Domain Forwarding**

// zone section fragment of named.conf

zone "example.com" IN {

type forward;

forwarders {10.0.0.1; 10.0.0.2;};

};

Where dial-up links are used with DNS forwarding servers BIND's general purpose nature and strict standards adherence may not make it an optimal solution. A number of the [Alternate DNS solutions](http://www.zytrax.com/books/dns/apc/) specifically target support for such links. BIND provides two parameters dialup and heartbeat-interval (neither of which is currently supported by BIND 9) as well as a number of others which can be used to minimise connection time.

Example configuration files for a forwarding DNS [are provided](http://www.zytrax.com/books/dns/ch6/index.html#forwarding).

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.5 Stealth (a.k.a. DMZ or Hidden Master) Name Server**

A stealth server is defined as being a name server which does not appear in any **publicly visible** [NS Records](http://www.zytrax.com/books/dns/ch8/ns.html) for the domain. The stealth server can be roughly defined as having the following characteristics:

1. The organisation needs a public DNS to enable access to its public services e.g. web, mail ftp etc..
2. The organisation does not want the world to see any of its internal hosts either by interrogation (query or zone transfer) or should the DNS service be compromised.

A Stealth configuration is shown in Figure 4-5.

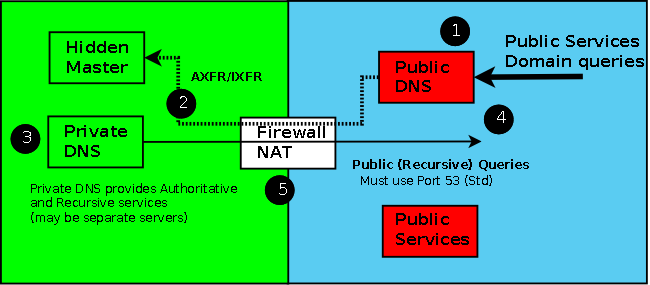


Figure 4-5 Stealth Server Topology

The external server(s) is(are) configured to provide [Authoritative Only](http://www.zytrax.com/books/dns/ch4/#authoritative) responses and no caching (no recursive queries accepted). The zone file for this server would be unique and would contain ONLY those systems or services that are publicly visible e.g. SOA, NS records for the public (not stealth) name servers, MX record(s) for mail servers and www and ftp service A records. Zone transfers can be allowed between the public servers as required but they MUST NOT transfer or accept transfers from the Stealth server. While this may seem to create more work, the concern is that should the host running the external service be compromised then inspection of the named.conf or zone files must provide no more information than is already publically visible. If 'master', 'allow-notify','allow-transfer' options are present in named.conf (each of which will contain a private IP) then the attacker has gained more knowledge about the organisation - they have penetrated the 'veil of privacy'.

There are a number of articles which suggest that the [view statement](http://www.zytrax.com/books/dns/ch7/view.html) may be used to provide similar functionality using a single server but this does not address the problem of the DNS host system being compromised and by simple inspection of the named.conf file additional data about the organisation could be discovered. In our opinion 'view' does not provide adequate security in a 'Split DNS' solution.

A minimal public zone file is shown below:

; public zone master file

; provides minimal public visibility of external services

example.com. IN SOA ns.example.com. root.example.com. (

2003080800 ; se = serial number

3h ; ref = refresh

15m ; ret = update retry

3w ; ex = expiry

3h ; min = minimum

)

IN NS ns1.example.com.

IN NS ns2.example.com.

IN MX 10 mail.example.com.

ns1 IN A 192.168.254.1

ns2 IN A 192.168.254.2

mail IN A 192.168.254.3

www IN A 192.168.254.4

ftp IN A 192.168.254.5

The internal server (the Stealth Server) can be configured to make visible internal and external services, provide recursive queries and all manner of other services. This server would use a private zone master file which could look like this:

; private zone master file used by stealth server(s)

; provides public and private services and hosts

example.com. IN SOA ns.example.com. root.example.com. (

2003080800 ; se = serial number

3h ; ref = refresh

15m ; ret = update retry

3w ; ex = expiry

3h ; min = minimum

)

IN NS ns1.example.com.

IN NS ns2.example.com.

IN MX 10 mail.example.com.

; public hosts

ns1 IN A 192.168.254.1

ns2 IN A 192.168.254.2

mail IN A 192.168.254.3

www IN A 192.168.254.4

ftp IN A 192.168.254.5

; private hosts

joe IN A 192.168.254.6

bill IN A 192.168.254.7

fred IN A 192.168.254.8

....

accounting IN A 192.168.254.28

payroll IN A 192.168.254.29

Using BIND 9's [view](http://www.zytrax.com/books/dns/ch7/view.html) statement can provide different services to internal and external requests can reduce further the Stealth server's visibility e.g. forwarding all DNS internal requests to the external server.

Example configuration files for a stealth DNS [are provided](http://www.zytrax.com/books/dns/ch6/index.html#stealth).

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.6 Authoritative Only Server**

The term **Authoritative Only** is normally used to describe two concepts:

1. The server will deliver Authoritative Responses - it is a zone master or slave for one or more domains.
2. The server will NOT cache.

There are two configurations in which Authoritative Only servers are typically used:

1. As the public or external server in a [Stealth (a.k.a. DMZ or Hidden Master) DNS](http://www.zytrax.com/books/dns/ch4/#stealth) used to provide perimeter security.
2. High Performance DNS servers. In this context general purpose DNS servers such as BIND may not provide an ideal solution and there are a number of [Open Source Alternatives](http://www.zytrax.com/books/dns/apc/) some of which specialise in high performance Authoritative only solutions.

You cannot completely turn off caching in BIND but you can control it and provide the functionality described above by simply turning off recursion in the 'option' section of named.conf as shown in the example below.

// options section fragment of named.conf

// recursion no = limits caching

options {

directory "/var/named";

version "not currently available";

recursion no;

};

// zone file sections

....

BIND provides three more parameters to control caching, [max-cache-size](http://www.zytrax.com/books/dns/ch7/queries.html#max-cache-size) and [max-cache-ttl](http://www.zytrax.com/books/dns/ch7/queries.html#max-cache-ttl) neither of which will have much effect on performance in this particular case and [allow-recursion](http://www.zytrax.com/books/dns/ch7/queries.html#allow-recursion) which uses a list of hosts that are permitted to use recursion (all others are not).

Example configuration files for a authoritative-only DNS [are provided](http://www.zytrax.com/books/dns/ch6/index.html#authoritative).

[up icon](http://www.zytrax.com/books/dns/ch4/#contents)

**4.7 Split Horizon DNS Server**

This section was introduced at the suggestion of Maren Leizaola - many thanks for both taking the time to write and for providing interesting usage examples.

The term **Split Horizon** is normally used to describe a DNS server that will give different responses (IP addresses) based on the source address, or some other characteristic, of the query. While it has similar configuration properties to the [Stealth Server](http://www.zytrax.com/books/dns/ch4/#stealth) it can also be used in a varity of unique situations such as:

1. **Geographic Mapping:** Assume that, for example, a web service is replicated in a number of locations (for either performance or access latency reasons) then a specific IP address may be returned based on the source address of the query to ensure the shortest possible path from the user to the service. For those familiar with **anycast** you could consider this as a poor man's anycast service.
2. **Naming Consistency:** Assume that you have, say, a corporate in-house LDAP service and that you want to keep certain highly secure data on one server only accessible to certain individuals or organizational sections, which have unique or identifiable IP addresses or address ranges, but for reasons of consistency (scripts, configuration files etc) you want both the secure and insecure LDAP services to be named, say, ldap.example.com.
3. **Load Balancing:** Assume that an analysis of incoming service users shows that their source-ip addresses can be separated into contiguous ranges: 50% from a to b, 50% from b to c. In this case rather than simply provide multiple A/AAAA RRs (where load balancing is essentially random) it may be more effective to use a split-horizon strategy.

Other possibilities may strike imaginative readers. The unifying element is that some characteristic of the incoming query will cause the DNS to generate a query-dependent result.

BIND's [view](http://www.zytrax.com/books/dns/ch7/view.html) clause provides a method that can be used to build such configurations and example files [are provided](http://www.zytrax.com/books/dns/ch6/index.html#split-view) .