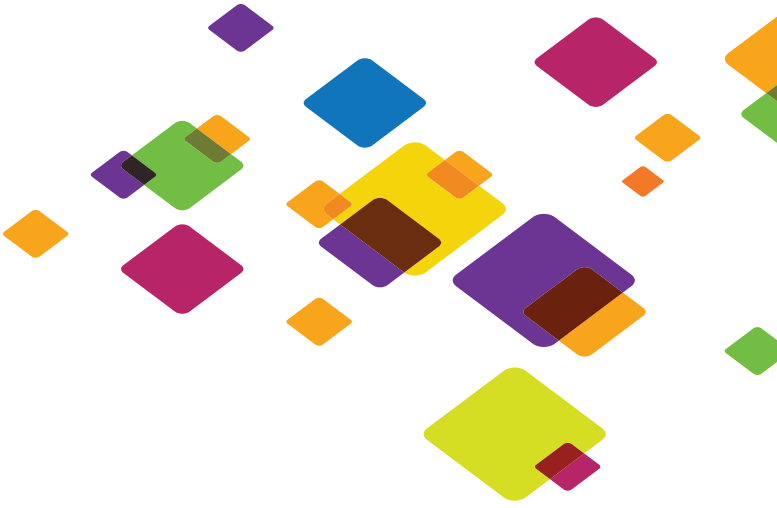
Infoblox
CONTROL YOUR NETWORK

WHITEPAPER

Disaster Preparedness for Core Network Services

Resiliency and Control for Disaster Recovery Planning and Business Continuity

Cricket Liu, Vice President of Architecture



Abstract

Core network services, which include name resolution (DNS), IP address assignment (DHCP) and management (IPAM), are used by just about every networked application, which these days includes pretty much every non-trivial application. This is not surprising: In the case of DNS alone, the richness and flexibility of modern networks and applications would simply not be possible if applications couldn't refer to devices and resources by symbolic names rather than IP addresses.

In this white paper, we'll take a look at how failures can unexpectedly disrupt core network services, such as name resolution and IP address assignment, and the applications that depend on them, despite employing traditional approaches to providing redundant configurations. Then we'll examine how we can build our service infrastructure to be as resilient as possible, to minimize the effects of failures and ensure business continuity.

A Cautionary Tale

Unfortunately, I have first-hand experience with unexpected disruption of services. In 1989, I was working for Hewlett-Packard at the company's corporate offices in Palo Alto, California. During the Loma Prieta earthquake, a sprinkler main that ran above one of HP's computer rooms cracked, flooding the room. And while HP computers are reliable, most aren't designed to operate while submerged. One of the computers we lost in the deluge was the primary hp.com name server.

HP's DNS infrastructure was designed according to industry best practices at the time. We had a primary name server on which we managed the data in the hp.com zone, and we had many secondary name servers distributed throughout the company that got their data from the primary and responded to queries from resolvers and other name servers. When disaster struck, we knew we had some time before the secondaries would stop responding to queries in hp.com (the hp.com zone's expire interval, to be precise). But without an operational primary name server, we couldn't modify the hp.com zone in any way. That was unacceptable: HP depended on our ability to modify the records of hosts that, like the hp.com primary, had been lost to flooding; to add new hosts; and to fix problems with delegation to subdomains of hp.com. So a friend and I spent a long, unpleasant night restoring the primary's function onto different hardware in a different data center—a painful lesson in manual disaster recovery.

After we'd configured a new name server as the primary for hp.com—even giving it the same IP address as the former primary to obviate reconfiguring the secondaries for hp.com—we had only won half the battle. We could make manual changes to the hp.com zone, but the complex process that generated the zone data from the internal host table and delegation information depended on a system that was literally under water. It took several days to re-establish full operations.

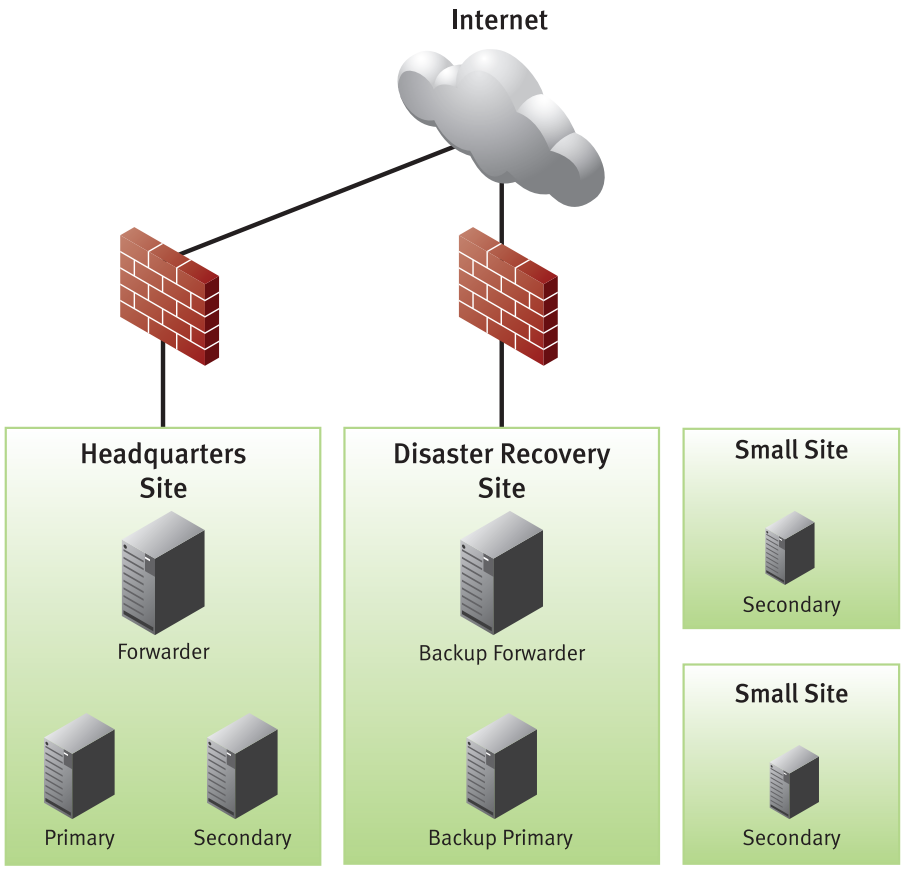
Today, given the company's dramatically increased reliance on DNS, requiring several days to restore management of the zone would be unthinkable, as it would literally cripple a global IT operation. But many large companies are in situations similar to HP's in 1989, whether they're aware of it or not. And, should their systems fail entirely, the repercussions would be much more severe, as core network services have become critical with the continued growth in IP-based devices and applications.

Common Misconceptions about DNS and DHCP Compromise Service Availability

In the case of DNS, conventional wisdom holds that the DNS protocol is inherently resilient; that when deployed correctly, failures in DNS servers or intermediate network components need not have a significant or even noticeable impact on network services and applications. To a certain extent, this is true: For example, you can configure a DNS client (a "stub resolver") to query multiple name servers so that the failure of just one of those name servers won't cripple the resolver's ability to resolve domain names. You can configure a name server to use multiple forwarders, so that if a single forwarder crashes, your name server will have other forwarders to query. These capabilities support the notion that resiliency is inherent in the design of the Domain Name System.

This conventional wisdom, unfortunately, ignores other, less obvious consequences of failures that can still cause severe problems. If the first name server a resolver is configured to query doesn't respond, for example, the resolver will wait for a timeout before trying the next name server in the list. Many resolvers don't change their behavior when the first name server fails to respond, and continue to query it first any time they have a domain name to resolve, thereby incurring a timeout for each reach request. This translates to delays and, in many cases, degraded application performance. An application that sends lots of DNS queries—a mail transport agent, or a complex financial application, for example—might slow to the point that it's unusable or fail completely due to excessive timeouts, even though DNS is technically still available.

Here's an example of a network and service-delivery architecture fairly common among large companies today:



The network has two major sites: a headquarters site and a disaster recovery (DR) site. Both the headquarters and the disaster recovery site have data centers with connections to the Internet. Smaller sites connect to both data centers for redundancy. Most critical name servers in the headquarters data center—the forwarder, the primary name server—have counterparts in the disaster recovery data center.

At the headquarters site, resolvers are configured to query both the local primary and secondary, ordered according to proximity (i.e., the closer name server is queried first). The primary and secondary name servers at headquarters also run DHCP servers. DHCP scopes for the headquarters site are split between the DHCP servers, with 80% of leases managed by one and 20% by the other.

Smaller sites run servers that support both DNS and DHCP. These internal name servers are configured to query both the main and DR forwarders, and are configured to use both the primary at headquarters and the "backup primary" at the disaster recovery site as masters for secondary zones. Stub resolvers are configured to query a "local" name server first, then to fall back to a name server at headquarters, then to a name server at the disaster recovery site. DHCP scopes are split between DHCP servers at smaller sites and a central DHCP server at headquarters, with 80% of leases managed by the local DHCP server and 20% by the headquarters DHCP server.

Examining the Failure of Remote DNS and DHCP Servers

Let's examine how the failures of various components would affect delivery of DNS and DHCP. To begin, we'll consider the failure of a remote DNS/DHCP server at a small site. In some organizations, this could be a relatively commonplace occurrence: With many remote name servers to administer, each possibly sharing a hardware and operating system platform with other services, ensuring stability is a challenge.

Stub resolvers that are configured to query this remote name server first (e.g., resolvers at the same site) will suffer some degradation in performance. BIND resolvers, standard in Unix and Linux operating systems, always query name servers in the same order, and hence their first queries will always time out, causing a two- to five-second delay for the application that requested the resolution. Modern Windows resolvers may dynamically reorder their queries, but will still incur some penalty when they first try the "down" name server.

More significant is the loss of local DHCP service. Since the company uses split scopes, the "backup" DHCP server at headquarters can continue to give out leases to clients from its smaller lease pool. However, the useable address space is reduced to 20% of its normal size, so unless the total number of leases in use is less than 20% of the total allocated, at some point clients won't be able to get IP addresses when they try to renew, or as new devices attempt to join the local network. Even those DHCP clients lucky enough to get a lease will get a different IP address. Many devices, particularly VoIP phones, don't react well when they can't renew their IP addresses: Receiving a new IP address in a lease means dropping calls.

Also consider the amount of effort required to restore the remote DNS and DHCP server. Even if we ignore any other services the host might have been running, restoring the DNS and DHCP services may require loading the right operating system on replacement hardware, downloading and installing various service packs and

patches, securing the operating system, loading name server and DHCP server software, restoring DNS and DHCP configurations from backup media or another host, testing the resulting configuration, and getting it installed at the remote site. This process will require several hours at least, and depending on the situation could take several days, during which many users at the remote site would have to make do with degraded network connectivity.

Failure of the Headquarters Primary

Now let's consider the failure of the primary name server at the headquarters site. This is, after all, a contingency the company has planned for: They've gone to the trouble of establishing a backup primary at their disaster recovery site. You'd hope that such a failure wouldn't cause a major outage.

At the very least, the backup primary is configured as a secondary for all of the zones on the primary. This will give it a reasonably up-to-date copy of the zone data. The company has probably also implemented some additional mechanism to synchronize other files between the headquarters primary and the backup primary. This might involve using a file synchronization tool such as rsync to copy name server configurations (e.g., named.conf files) from the headquarters primary to the backup, for example. If the administrative infrastructure on the headquarters primary is more complicated, of course, the synchronization will have to be more sophisticated. For example, if the company uses a conventional IP address management (IPAM) product, they may need a more sophisticated synchronization system, such as a commercial mirroring product.

No matter how sophisticated the synchronization, however, some loss of data is nearly inevitable when a failure occurs. If the synchronization runs only periodically—say, every 8 hours–the backup will only have data as current as the latest sync. The synchronization mechanism may lose important dynamic updates, which are only written to disk some time after they're received. And with a conventional IPAM system, the data in the headquarters system will be an outdated snapshot of what was actually on the primary, secondary and remote servers at the time of the failure.

Restoring the primary at headquarters is also a headache. As with restoring a remote name server, this may require loading an operating system, downloading and applying service packs and patches, securing the operating system, and so on. Once this is complete, the synchronization between primaries must be performed again, but in the reverse direction: Any changes made on the backup primary must be synchronized with the headquarters primary. And again, some data, including dynamic updates that haven't yet been written to disk, may be lost.

Configuration Error on the Headquarters Primary

Imagine an administrator making a configuration error on the primary name server at the headquarters site. (This should be easy to imagine, given how common it is.) Maybe this is just a small syntax error in a zone data file or named.conf file. Unfortunately, many name servers—BIND name servers in particular—are notoriously unforgiving of seemingly minor syntax errors. Depending on the make and model of name server software, this could cause the primary name server not to restart, to begin issuing SERVFAIL responses to queries, or any of a number of other unappetizing possibilities.

Since some of the resolvers at the headquarters site are configured to query the primary first (specifically, those resolvers closer to the primary than to the secondary), the error will have immediate effects. Resolvers querying the primary will begin returning any SERVFAIL errors to the applications that called them. If the primary didn't restart, resolvers will time out when querying it. After a few seconds, they'll query the secondary, but these timeouts will add up for software that sends lots of queries, such as mail servers.

Failure of the Headquarters Forwarder

One would hope that a failure of the forwarder at the headquarters site wouldn't cause undue problems. As with the backup primary, the company has gone to the trouble and expense of implementing a backup forwarder at the disaster recovery site. Unfortunately, because of limitations in name server implementations, even the failure of a single forwarder can have far-reaching implications.

Most name servers that are configured to query multiple forwarders simply query the first forwarder in the list, and if it fails to respond within a timeout (typically a few seconds) it will try the next forwarder in the list. For example, when configured with this options statement, most BIND name servers would first query the forwarder at 10.0.0.1 and, if it didn't respond within a few seconds, would query the forwarder at 10.0.1.1:

```
options {
        forwarders { 10.0.0.1; 10.0.1.1; };
};
```

At first glance, this behavior seems sensible enough. However, if all internal name servers are configured this way, and the first forwarder in the list fails, all internal name servers will incur a several-second timeout for each forwarded query. Busy name servers handle hundreds or thousands of queries per second, and will quickly develop a backlog of recursive queries awaiting an answer from a forwarder. By default, once this backlog reaches 1000 recursive queries, a BIND 9 name server will refuse additional recursive queries. Other name servers may not impose a hard limit on concurrent recursive queries but may run out of memory or other resources. Either way, the failure of the single forwarder can cascade into an inadvertent denial of service to clients throughout the network.

Newer BIND name servers, from BIND 9.3.0 forward, choose among forwarders according to roundtrip time instead of simplemindedly walking the list. This helps reduce the impact of losing a forwarder considerably. However, many companies are still running older BIND name servers internally, or Microsoft DNS Servers, which also lack an intelligent algorithm for selecting forwarders and are therefore prone to this type of DNS problem.

Best Practices for Building Resilient DNS and DHCP Infrastructure

Companies have many options for mitigating the impact of failures like these.

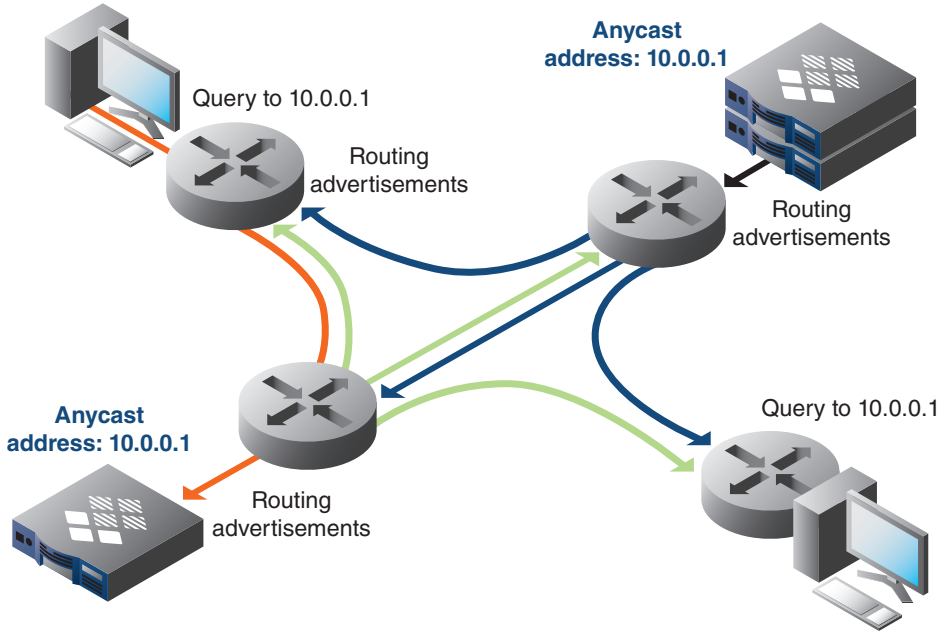
Failure of a Remote DNS and DHCP Server

The most obvious way to mitigate the failure of a DNS/DHCP server is to implement a high-availability (HA) configuration. Unfortunately, providing HA for conventional operating systems and applications is a non-trivial exercise. It typically requires additional software products (e.g., Microsoft Clustering Services), which can be fairly complex to set up and manage, and may not be able to guarantee perfect synchronization of data between servers in the cluster.

Appliances that support VRRP-based high availability, allowing administrators to deploy critical name servers and DHCP servers as high availability pairs, can also address this problem. The two appliances in an HA pair share a virtual IP address. At any one time, only the active member of the pair uses the virtual IP address. Any changes to protocol configuration or data on the active appliance—including changes to DHCP lease state and dynamic updates—are replicated to the passive member. This ensures that no dynamic updates are lost and that no duplicate IP leases will be issued by the backup appliance. If the active member fails, the passive member becomes active and takes over the virtual IP address—a process that usually requires only two to five seconds. The members of an HA pair can even reside in different sites, as long as a VLAN is extended between them.

For those who can't justify a pair of servers at a remote site, a good way to minimize the disruption caused by the failure of a remote name server is to use a technique called "anycast." In anycast configurations, many name servers share a single virtual IP address. The hosts that run these name servers advertise routes to this IP address to their neighbor routers, and those routers direct queries sent to the virtual IP address to the closest name server. When a name server fails, its host notices and withdraws the route, causing the routers to recalculate the routing table and send queries elsewhere.

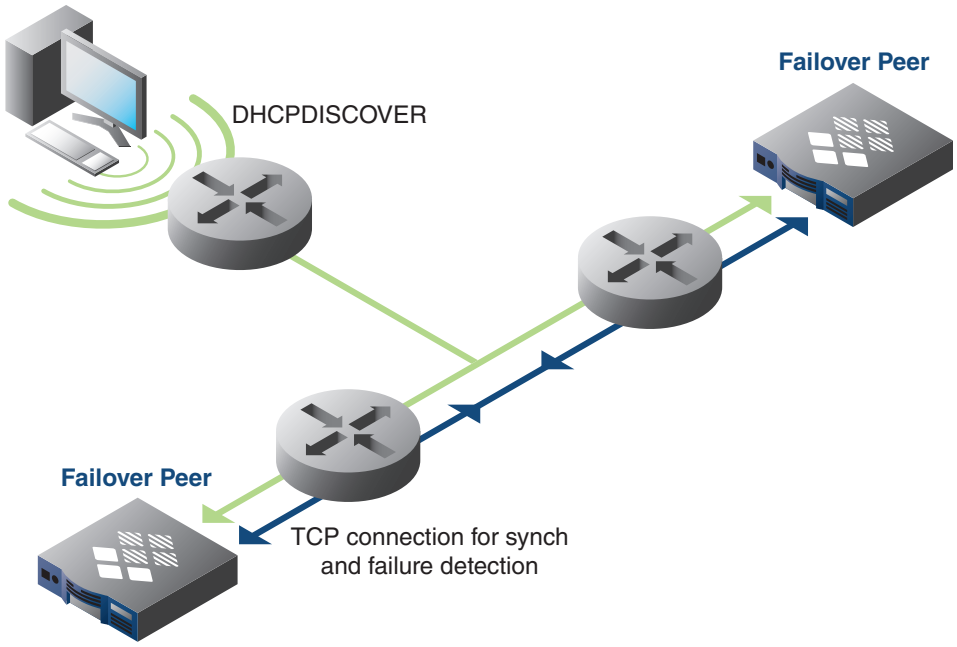
Using anycast addressing for name servers simplifies resolver configuration considerably: Instead of configuring the resolver with a list of name servers to query, you may only need one. Instead of customizing configurations for the resolvers at each site, you can use the same resolver configuration everywhere. And when a name server fails, resolvers don't suffer a multi-second timeout before trying the next name server in the list.



The trouble with anycast configurations is that they're difficult to set up. Anycast support isn't built in to most operating systems, so administrators are forced to install and configure their own routing daemon on each name server host, and to develop custom code to monitor the name server and adjust the routing advertisements accordingly.

In some appliances, anycast support for DNS is a standard part of the product. All that's needed to enable it is to assign a virtual IP address to the appliance and to configure the embedded routing daemon to communicate with its neighbors.

Unfortunately, anycast and DHCP don't work well together. There's no standard mechanism for synchronizing the lease database of an arbitrary number of DHCP servers serving the same lease pool. However, DHCP Failover lets you synchronize a lease database between two DHCP servers. With DHCP Failover, a lease pool is assigned to two different DHCP servers, which are configured in what's called a failover association. Each synchronizes its lease activity with the other, reporting which IP address it's leased to which client and for how long. If either peer in the association should fail, the other can take over issuing and even renewing leases for the other. This eliminates the problems that arise using split scopes when the backup server hands out new addresses to renewing clients (like IP phones) and also avoids the problem of the smaller backup scope becoming exhausted as clients from the larger, main lease pool renew their leases.

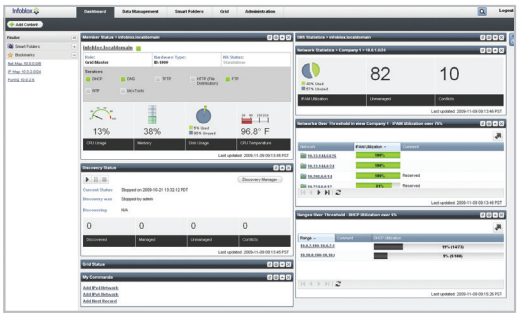


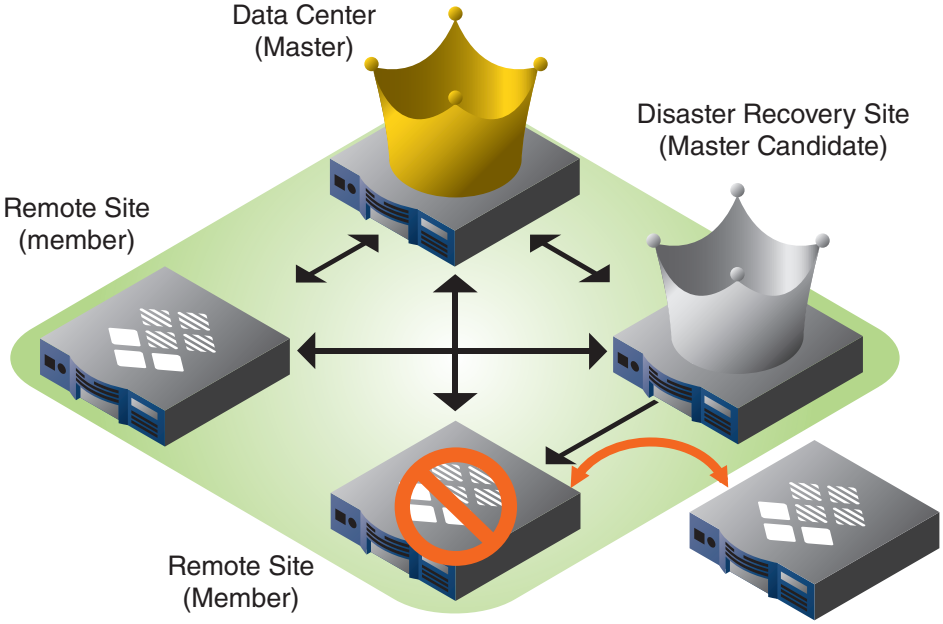
While recent versions of ISC's DHCP server include support for DHCP Failover, the Internet Draft that describes the mechanism has serious shortcomings. Making administrative changes to DHCP servers in a failover association can remove usable IP addresses from a lease pool. External factors can cause DHCP servers in a failover association to enter states from which they can't recover without a restart. Look for appliances with DHCP Failover implementations that have been extended to address these issues and provide DHCP Failover that works consistently and reliably in real-world conditions.

Finally, there are techniques that will reduce the amount of time and effort necessary to restore a remote DNS and DHCP server. Creating a standard operating system image, with all necessary service packs and patches applied, will ease recovery of the operating system on a general- purpose server. Backing up all name server and DHCP server configuration files and zone data files, even if only by copying them to another host, will ensure that you can recover basic service configuration. However, this "snapshot" approach can result in data loss due to discrepancies between data backed up and the real state of the server just before it failed.

Infoblox has developed unique technology that enables a collection of appliances to perform and be managed as a single unified system called a Grid. With appliances in an Infoblox Grid, recovery of a failed appliance is as simple as connecting a replacement appliance to the network and assigning it the IP address and basic network configuration of the failed unit. All of the failed appliance's configuration, including administrative settings, DNS and DHCP configuration—even dynamic updates—is stored in the Infoblox Grid, and replicated to the replacement appliance automatically, so that within minutes it can resume operation where its predecessor left off.

Infoblox Grid Manager

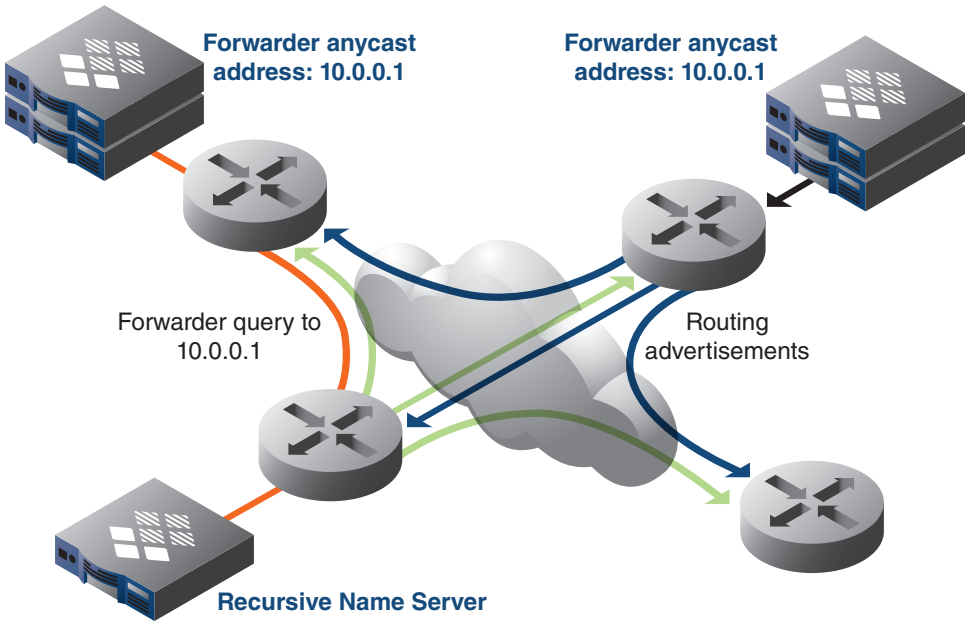




Failure of the Headquarters Forwarder

To prepare for the possibility of the loss of a forwarder, the company could front-end a pair (or more) of forwarders with a load balancer. The load balancer could manage a virtual IP address, which it then dynamically translates into the address of one of the

forwarders. Internal name servers offering recursion would be configured to forward queries to this virtual IP address. However, the load balancer itself becomes a single point of failure: If the load balancer crashes, internal name servers lose both of their forwarders! To deal with this problem, many load balancers themselves support high availability. At that point, though, the solution becomes awfully expensive.



Another option is to use anycast to distribute queries forwarded to the forwarders' virtual IP address. But this suffers from the disadvantage we cited earlier: complicated setup. Finally, the company could use appliances with built-in VRRP-based high availability and anycast as forwarders. For example, the company might run forwarders at the headquarters site and at the disaster recovery site that share the same virtual IP address. If one forwarder fails, the routing infrastructure will direct queries sent to the virtual IP address to the other forwarder.

Failure of the Headquarters Primary

When a primary name server fails, it's critical to avoid losing data and to restore the ability to manage DNS data as quickly as possible. This is especially true in a real disaster because DNS is an invaluable tool for redirecting clients and applications away from systems that have become damaged or unreachable and pointing them at established (or jury-rigged) backup resources. By backing up both the name server's configuration and zone data after every significant change, you reduce both the chance of data loss and the time to recover the primary. The backup mechanism could be as simple as using rsync to copy all important files to a remote site.

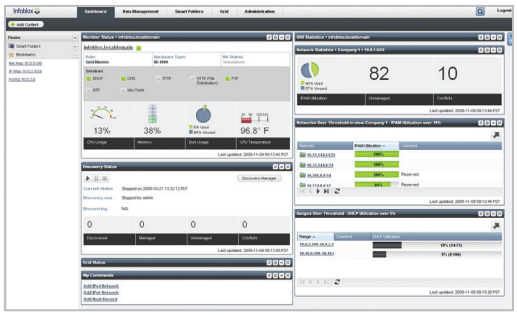
However, such a system ignores dynamic updates, which may be equally important. If a new Windows server has just registered DNS forward and reverse mappings for the services that it provides, but the corresponding records haven't yet been written to disk, a purely disk-based backup of the name server won't include that data. If that backup is used to restore the primary name server, the dynamic data will disappear, and applications pointed at the Microsoft server will fail. Preparing an operating system image, as described in the last section, can minimize the amount of time required to restore basic DNS and DHCP functionality. But even with an OS image, bringing up a replacement server, loading an operating system and re-establishing the primary name server can take hours.

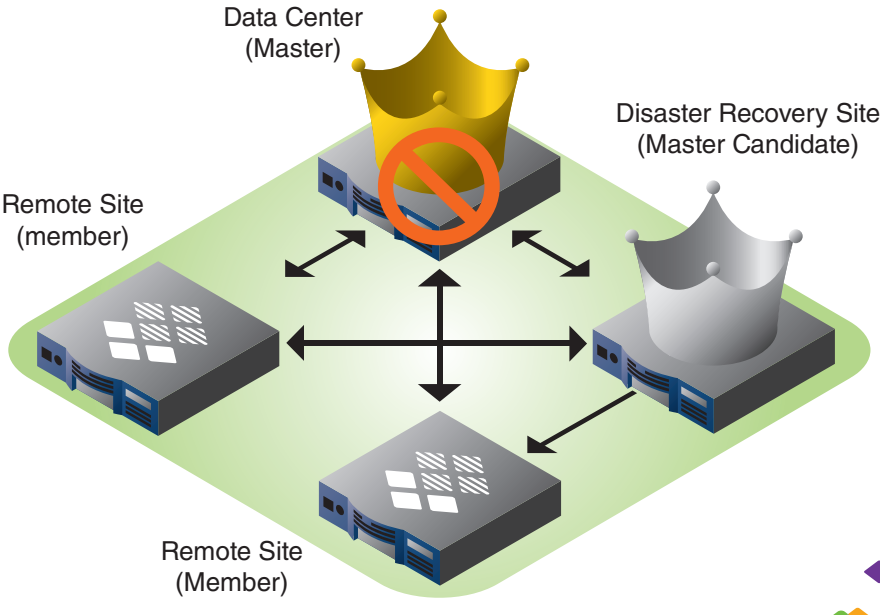
Companies that use conventional IP address management systems face a more daunting challenge, since most IPAM systems require a central database to store DNS and DHCP configuration and data. Consequently, recovery involves not just loading the operating system and restoring backed up data, but also loading and configuring database software. Not only does this require more time, but also specialized expertise.

For companies that maintain a hot DR site, the issue of lost data still remains. The data at the DR site will reflect the state of the network as it existed as of the most recent snapshot of the primary name server, DHCP server and IPAM system. Few companies have invested in sophisticated, real-time file and database synchronization tools for their DNS and DHCP systems, so of necessity some of the data at the DR site will be outdated and incorrect. As a result, even if the DR systems can be brought up quickly, the administrators will be left with the challenge of verifying the data and making manual changes to bring the DR primary name server, DHCP server and IPAM system in line with the actual network.

In an Infoblox grid, the rough equivalent to the server that runs the central database in a conventional IPAM system is the Grid Master. Unlike most traditional IPAM systems, though, the Grid Master is just another Infoblox appliance running the same software as all of the other appliances—not a dedicated, general-purpose server or specialized IPAM server. In most cases, the Grid Master is deployed on a high availability pair of appliances, since it plays a critical role within the Grid. In the event that both members of the HA pair are simultaneously incapacitated (for example, because of a loss of power at the headquarters site or segmentation of the network), any other appliance in the Grid that has been designated as a Grid Master Candidate can be promoted to the new Grid Master in single operation.

Infoblox Grid Manager





The Grid Master continuously replicates its database to the Grid Master Candidate, so the Grid Master Candidate's database reflects all data and configuration across the entire grid—including dynamic updates and leases handled by remote appliances. After promotion to Grid Master and synchronization of the grid—a process that usually takes only minutes—the appliances at the DR site support the ability to manage all DNS and DHCP data across the company. Providing this type of rapid recovery with minimal data loss is probably not possible without prohibitive investments in additional file and data replication and backup systems, extensive custom software development and complex manual recovery procedures.

Configuration Error on the Headquarters Primary

There are two ways to mitigate the risk of configuration errors on the primary name server. The first is to minimize the possibility of introducing an error. This could be done through the use of a graphical user interface to manage zone data and make configuration changes to the name server. A good GUI will check input fields for syntactic correctness and flag potential errors. It may also compartmentalize access to the name server's configuration, allowing administrators to manage only some of the zones on the name server and not, for example, the name server's forwarding configuration or access controls.

Companies that don't want to invest in a GUI or who simply prefer the command line can use utilities included in the latest versions of BIND to check the syntax of the zone data files and named.conf files before putting them into production. BIND 9's named-checkzone program checks the syntax of zone data files, while the named-checkconf program checks the syntax of named.conf files.

The other way of handling configuration errors is to minimize the damage they cause. A syntax error on the primary causes an immediate problem only because the company's resolvers and name servers query it directly, all the time. In a "hidden primary" configuration, the primary's sole function is to serve zone transfers to secondaries for the zone. Resolvers and remote name servers query the secondaries for information in the zone, not the primary. A syntax error on the primary won't propagate to the secondaries, who will refuse to transfer zones from the primary if the primary has any trouble loading them. Meanwhile, the secondaries continue responding to queries from resolvers and remote name servers with the last good zone data they transferred from the primary. They'll continue doing this until the zones' expire time passes, which is configurable, but usually at least two weeks.

Setting up a hidden primary configuration is easy: Simply leave out any NS records that refer to the primary, and configure resolvers to query the secondary instead. (You can even implement a query access control list on the primary, limiting name service only to the secondaries.) You might need to set up another secondary to assume some of the load previously served by the now-hidden primary, but that's a small price to pay for increased resiliency.

Summary

As “round-the-clock” business continuity becomes increasingly critical, organizations need to consider the importance of core network services to the continuous availability and, in the event of a disaster, recovery of every application on the network. To ensure the resiliency of your core network systems, consider these options:

- Implement an appliance-based infrastructure for core network services including DNS, DHCP, and IPAM
- Use a graphical user interface that includes built-in error checking to make all configuration changes
- Implement a “hidden primary” configuration to reduce the impact of configuration errors.
- Use high availability for critical DNS and DHCP servers with fast failover (VRRP) and data synchronization.
- When HA is too expensive, consider using anycast or DHCP Failover; make sure DHCP Failover implementations are “fixed.”
- Implement Infoblox appliances in a Grid to reduce time to recovery.
- Designate at least one Grid Master Candidate in your Infoblox grid to ensure your ability to recover from a disaster in minutes with no loss of DHCP, DNS or IPAM data.

If you continue to use software-on-servers rather than appliances:

- Make use of error-checking tools before committing changes
- Implement a “hidden primary” configuration to reduce the impact of configuration errors.
- Create a standard, pre-patched OS image to reduce time to recovery of failed servers.
- Implement a real-time database synchronization system to back up your IPAM server using real-time mirroring software. Define and practice procedures for restoring lost data (e.g. dynamic updates, leases) that are in the DNS and DHCP servers but not in the IPAM database at the time of failure.

CORPORATE HEADQUARTERS:

+1.408.986.4000

+1.866.463.6256

(toll-free, U.S. and Canada)

info@infoblox.com

www.infoblox.com

EMEA HEADQUARTERS:

+32.3.259.04.30

info-emea@infoblox.com

APAC HEADQUARTERS:

+852.3793.3428

sales-apac@infoblox.com