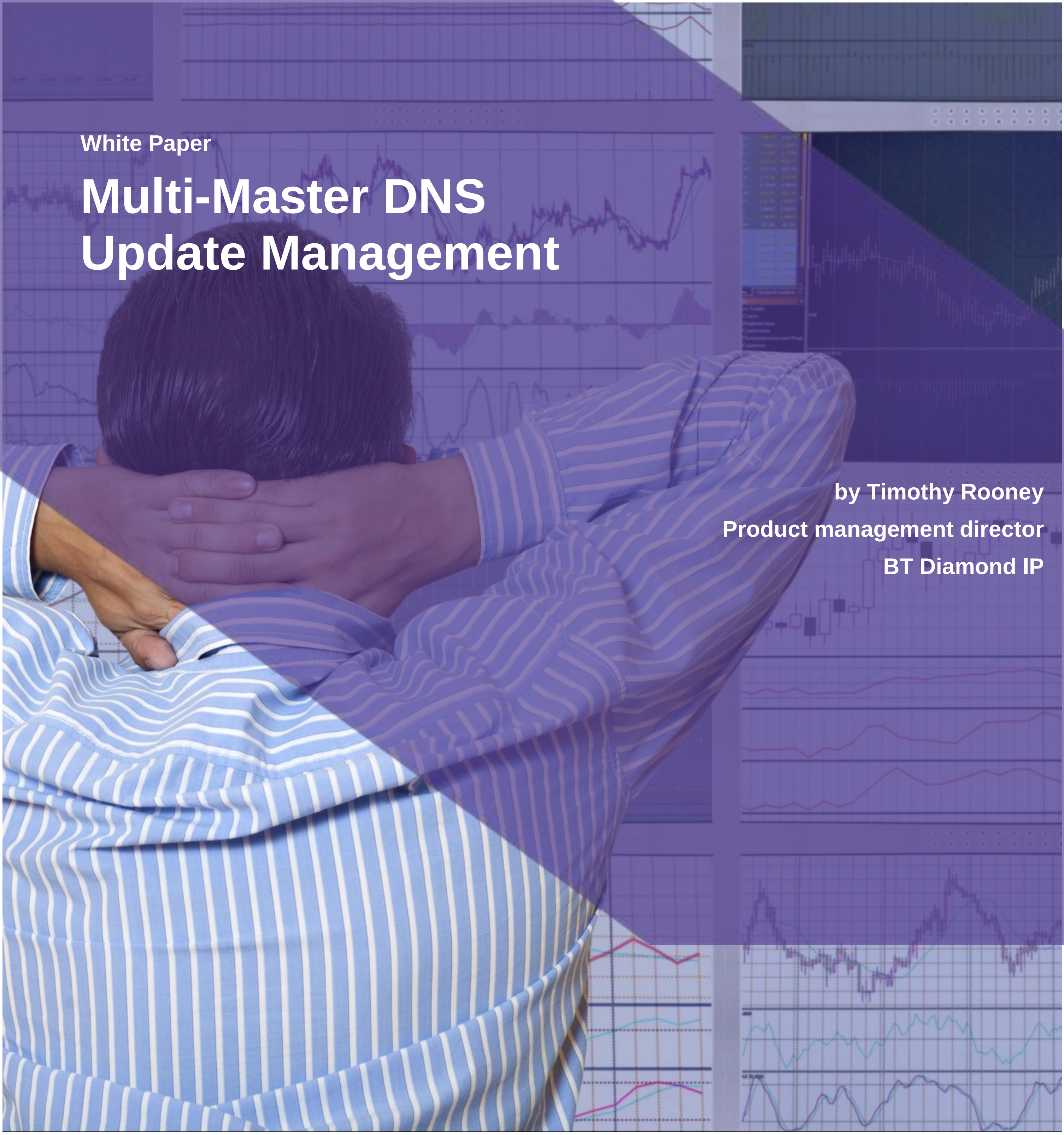
White Paper

Multi-Master DNS Update Management



by Timothy Rooney

Product management director

BT Diamond IP

Introduction

DNS is a foundational element of IP communications. To communicate over an IP network, an IP device needs to send IP packets to the intended destination; and each IP packet header requires both source and destination IP addresses. DNS provides the translation from a user-entered named destination, e.g., web site www address, to its IP address such that the sending device may populate the destination IP address with the address corresponding to the entered domain name. Thus, DNS provides the means for improved usability of IP applications by insulating end users from typing IP addresses directly into applications like web browsers and enabling web servers to serve web pages compromised of diverse linked content.

DNS is not only useful for Internet users but also for network administrators. By publishing name-to-IP address mappings in DNS, the administrator gains the freedom to change IP addresses as needed for network maintenance, changing providers, or general renumbering without affecting how end users connect. I can map my www address to 192.0.2.55 today and change it to 198.51.100.23 tomorrow without affecting how users reach my website. While you may not change your website IP address on a daily basis, devices which dynamically obtain IP addresses such as mobile devices may use different IP addresses on a regular basis. Others needing to connect to these dynamically-addressed devices rely on DNS resolving the device name with its current IP address.

Given its criticality in network communications, it's crucial that DNS services not only provide accurate information but that they be highly available such that DNS resolution services are available at all times. DNS architecture best practices for deploying highly available DNS services has been documented in our *DNS Architecture and Design Using IPControl* white paper by Alex Drescher. While that paper focused on highly available DNS resolution services, this white paper focuses on the dynamic nature of DNS resolution data and how DNS data changes can be efficiently and rapidly communicated. Such communications are necessary to maximize the likelihood that a given query for certain information is resolved in exactly the same way no matter which DNS server is queried.

DNS Change Management

DNS administrators are responsible for the DNS data published in their DNS servers. This data corresponds to a particular DNS domain within an internal or the Internet domain tree for which the administrator is authoritative. Resolvers internally or on the Internet will seek resolution data for this domain from servers to which you've published this authoritative data. The aforementioned DNS architecture white paper and Internet standard DNS practices call for deployment of authoritative data on multiple DNS servers, which enables the resolution of domain data reliably even in the face of a DNS server or DNS server link outage.

The main challenge then becomes how a given DNS data change from an accepted source is propagated to all DNS servers that are authoritative for that DNS data (i.e., configured with the corresponding zone data) as quickly and reliably as possible. Changes to DNS data can include additions, modifications, or deletions of DNS zones and servers, zone and server parameters, and DNS resource records. We'll examine the sources of these data changes, how the Internet-standard DNS architecture accounts for propagating changes and an alternative approach to provide rapid and reliable change management with resiliency.

DNS Data Change Sources

DNS data can be changed by administrators desiring to add, modify or delete DNS information to reflect network initiatives or projects. Most customers using an IP address management (IPAM) solution such as IPControl™ from BT Diamond IP can make changes as desired to DNS zones, servers and resource records, then deploy the changes automatically or on-demand. On-demand or scheduled changes provide a means for manual intervention, e.g., for formal change approvals.

These types of “top-down” changes are typically deterministic and can be planned, deployed and monitored through completion to assure timely change replication to all servers. However, a DNS change source which by nature provides a higher change frequency and from distributed sources is that of your Dynamic Host Configuration Protocol (DHCP) servers (or end clients if that’s permitted on your network). DHCP servers enable the pooling of IP addresses to enable mobile devices to obtain an IP address on a (renewable) temporary basis for use while present within the span of the corresponding subnet. When the device moves to another location, it may relinquish the IP address, or it may just time out, freeing the IP address to be assigned to a different client. DHCP provides an efficient and automated means for dynamic IP address assignment.

With each IP address assigned to a given device, the mapping of the IP address to a host domain name corresponding to the present device needs to be updated in DNS. Thus, your DHCP servers are likely the source of the largest quantity of your DNS changes, which may occur at a frequency of hundreds or thousands per minute. How can you architect your DNS to propagate these updates rapidly and accurately? Let’s first explore how the Internet standard method addresses this.

Multi-authoritative DNS

DNS was designed for redundancy and reliability. It was also designed to simplify administration by enabling such changes to be enacted on one authoritative DNS server, which would then propagate the change to other authoritative DNS servers. The server on which changes are made is referred to as the *master*, while other authoritative servers for the same zone information which obtain updates from the master are referred to as *slaves*.

Slaves are configured to poll the master periodically to assure they have the most up-to-date zone information. Within the DNS protocol, each slave server issues a DNS query for the zone’s start of authority (SOA) resource record from the master. Within the SOA record, the zone serial number provides a means to determine if zone data has changed. Each time a change is made to a zone resource record (add, change delete), the zone serial number is incremented. In this manner, when a slave which has the current serial number of value *x*, receives an SOA record from the master indicating a serial number of *x+n*, the slave can infer that it’s version of the zone is outdated and it needs to refresh its zone information.

The slave may request a zone transfer from the master to obtain the latest version of the zone. The polling interval by which each slave may poll the master is also defined within the SOA record as the refresh time. So a given resource record change may take up to the refresh time to propagate to each slave, which could be on the order of several minutes.

The timeliness of this process was greatly improved through the introduction of a notification mechanism whereby the master issues a DNS Notify message to each of its slaves soon after a zone update is made to proactively convey a zone change. Upon receipt of the DNS Notify message, a slave may request an incremental zone transfer (IXRF) by issuing an IXRF query to the master including its version of the zone SOA in the query. The slave’s SOA indicates the serial number of the zone for which the slave is up-to-date; thus the master may compile a list of resource record changes made to the zone since the version with that serial number and respond to the slave with the incremental updates. This provides a timely and efficient means for each slave to resynchronize the zone with the master server.

The IPControl™ system from BT Diamond IP is a centralized IP address management (IPAM) solution that enables administrators to manage IP address space, DHCP configurations and DNS configurations including resource records. IPControl enables deployment of configuration, zone and resource record information from its centralized database and it also provides for the updating of resource records based on dynamic address assignments (e.g., via DHCP) performed autonomously within the network. In this manner, the IPControl system supports a single pane of glass perspective on DNS zone configurations supporting both a top-down and bottom-up management model.

IPControl's DNS Listener service is configured to appear as a slave to the DNS master server deployed in the network. As such it can receive the DNS Notify message and perform IXFRs to retrieve network-assigned resource record changes to given DNS zones. Figure 1 illustrates this process and the multi-authoritative update process in general. When a DHCP server assigns an IP address to a host, let's say IP address 10.0.0.1 to `foo.example.com`, the DHCP server can be configured to update a master DNS server with the name-to-IP address mapping. The DHCP server uses a process termed dynamic DNS (DDNS) to send a DNS Update message to the DNS master. The master DNS server should be configured with access control lists (ACLs) and even transaction signatures to constrain the set of allowed update sources to those trusted such as corresponding DHCP servers.

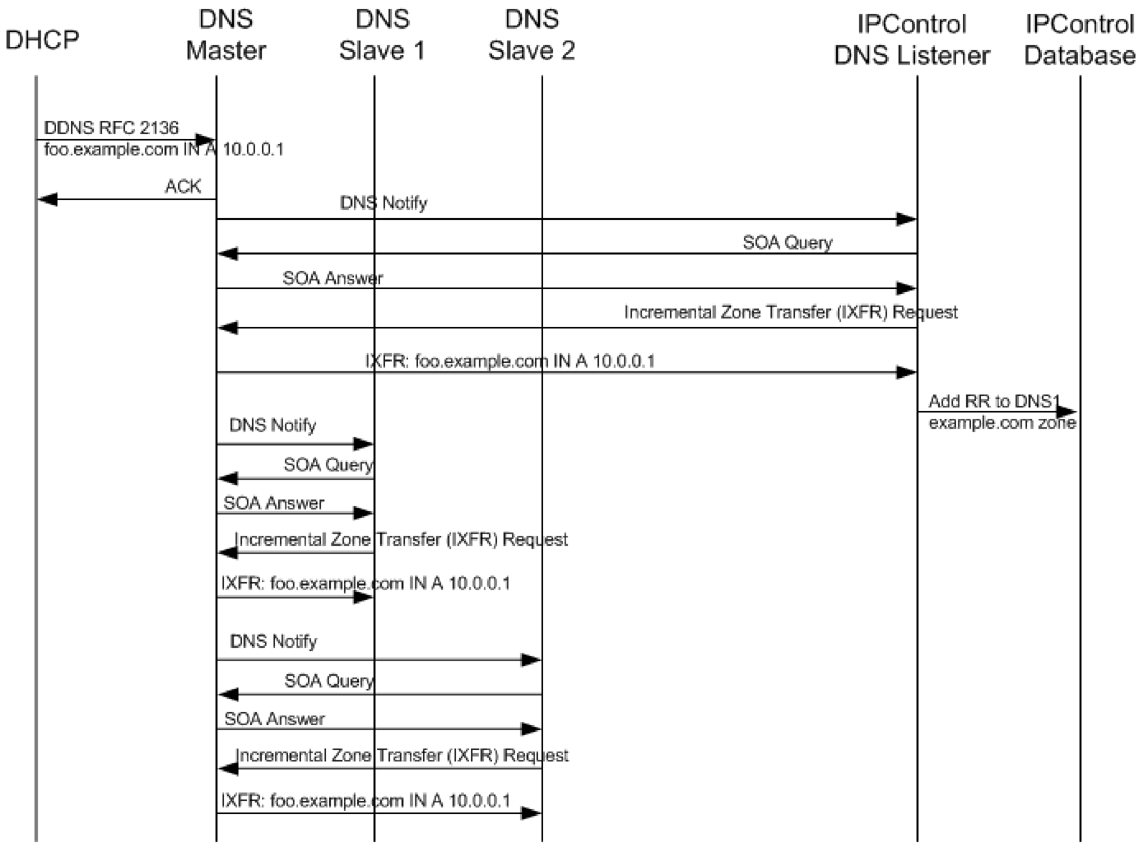

DHCP
DNS Master
DNS Slave 1
DNS Slave 2
IPControl DNS Listener
IPControl Database
DDNS RFC 2136
foo.example.com IN A 10.0.0.1
ACK
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request
IXFR: foo.example.com IN A 10.0.0.1
Add RR to DNS1
example.com zone
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request
IXFR: foo.example.com IN A 10.0.0.1
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request
IXFR: foo.example.com IN A 10.0.0.1

Figure 1: Multi-authoritative DNS update process

The DNS master will immediately or at a given interval issue DNS Notify messages to each slave server authoritative for the given zone, `example.com` in our case, including the IPControl DNS Listener. Each slave, including the DNS Listener would perform the IXFR to update its respective information for the zone.

This multi-authoritative approach offers the following advantages:

- Well-defined process for updating multiple DNS servers with updated zone resource record information
- Internet standard approach

This architecture however suffers from the following disadvantages:

- Updates cannot be made to the zone should the DNS master fail (single point of failure)
- Updates may take a few minutes to propagate to all authoritative servers during the Notify/IXFR process

Multi-tiered DNS

One of the main drawbacks of the multi-authoritative approach just described is its single point of failure with the single DNS master. If the DNS master server is unavailable, any resource record changes must be queued until the server is restored. One approach to enabling the processing of resource record updates while the master server is down entails promoting a slave to a master temporarily. The nominal case works just as in the single master case as shown in Figure 2 with the exception that non-primary slaves may request zone transfers from either the master or primary slave.

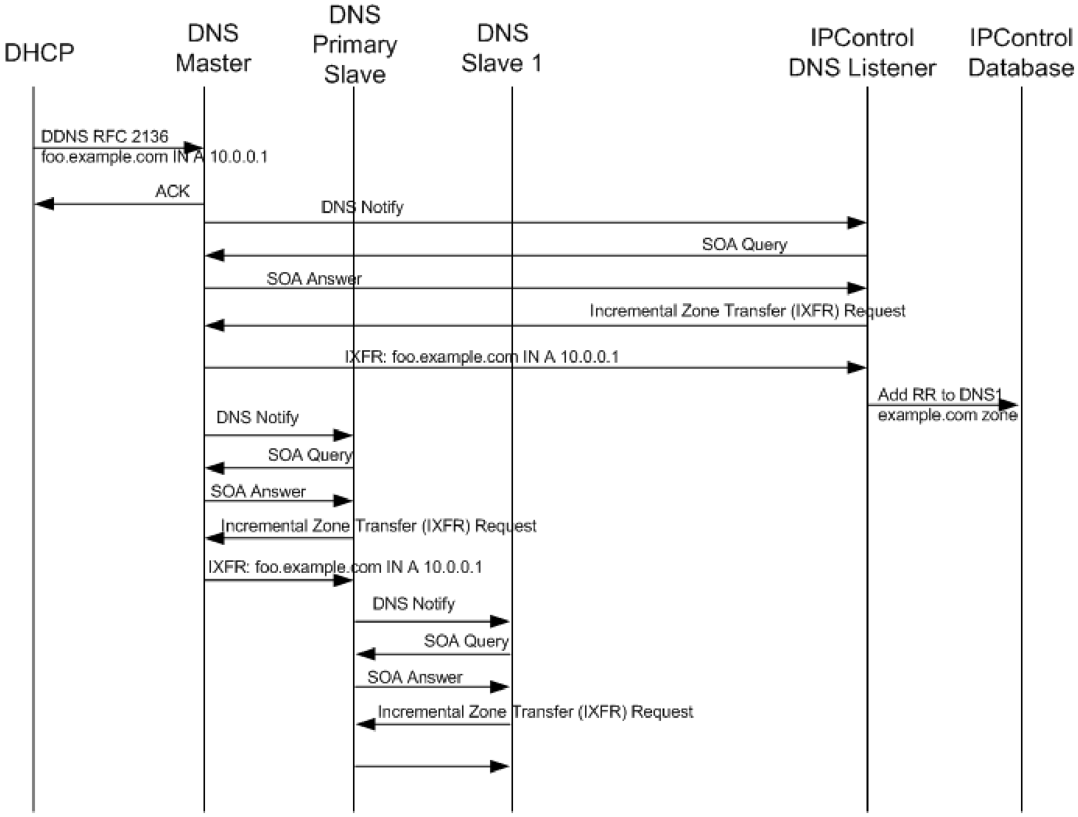

DHCP
DNS Master
DNS Primary Slave
DNS Slave 1
IPControl DNS Listener
IPControl Database
DDNS RFC 2136
foo.example.com IN A 10.0.0.1
ACK
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request
IXFR: foo.example.com IN A 10.0.0.1
Add RR to DNS1
example.com zone
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request
IXFR: foo.example.com IN A 10.0.0.1
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request

Figure 2: Multi-tiered DNS update architecture

When the master DNS fails, the primary slave must be “promoted” to the master. You can generally configure your DHCP server and your other slave servers for the zone with both masters’ IP addresses. Thus, when you promote the primary slave, you should not need to update the DHCP servers’ or other slaves’ configurations. However, you would need to modify your IPControl configuration to reconfigure the primary slave as master for the zone(s). Once this process has completed, the update process follows that outlined in Figure 3.

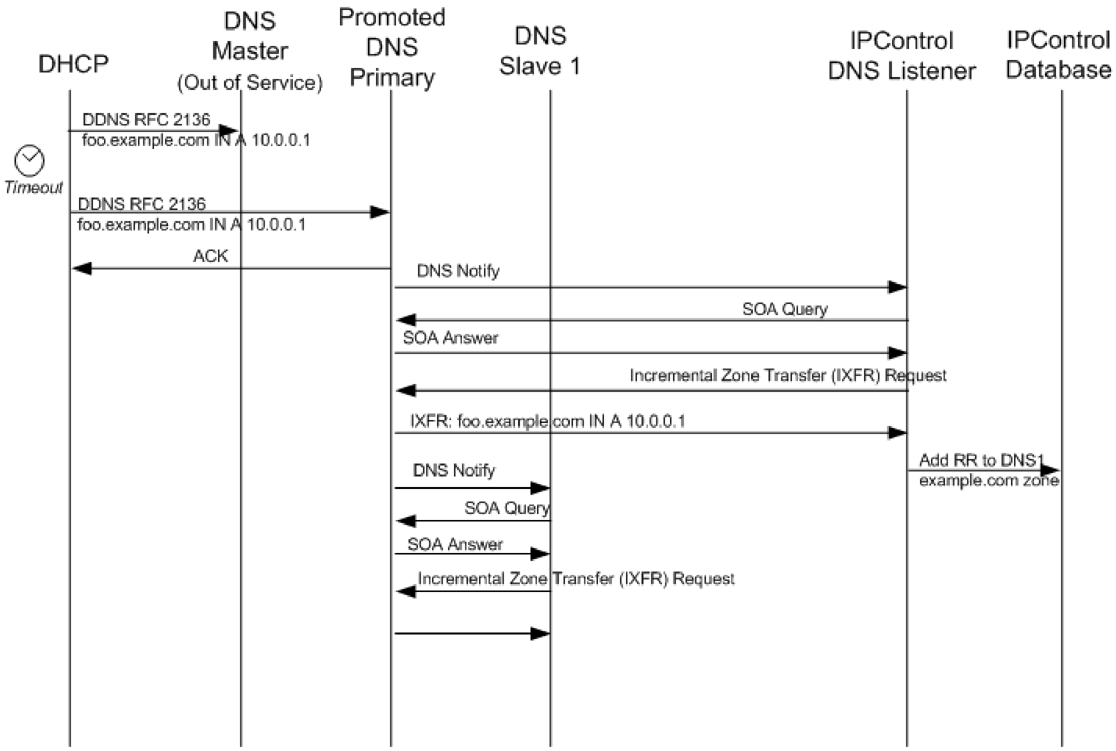

DHCP
DNS Master (Out of Service)
Promoted DNS Primary
DNS Slave 1
IPControl DNS Listener
IPControl Database
DDNS RFC 2136
foo.example.com IN A 10.0.0.1
Timeout
DDNS RFC 2136
foo.example.com IN A 10.0.0.1
ACK
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request
IXFR: foo.example.com IN A 10.0.0.1
Add RR to DNS1
example.com zone
DNS Notify
SOA Query
SOA Answer
Incremental Zone Transfer (IXFR) Request

Figure 3: Promoted DNS slave update process

Upon failing to receive an acknowledgement from the DNS master after attempting a DDNS update, the DHCP server would move to the second IP address in its configuration for the zone, which should point to the promoted DNS primary. This newly configured master will then perform the notify and IXFR processes to the IPControl DNS Listener and to other DNS slave servers.

The benefits of this multi-tiered approach include:

- Well-defined process for updating multiple DNS servers with updated zone resource record information
- Uses Internet standard protocols

This approach however suffers from the following disadvantages:

- Manual steps are required to promote a primary slave to a master
- Updates cannot be made while the master is down until the primary slave is promoted
- Updates may take a few minutes to propagate to all authoritative servers during the Notify/IXFR process after the primary slave has been promoted.

Multi-master DNS

The multi-master scenario implies the use of multiple DNS servers as master for a given zone at any one time. The primary benefits of this approach are that the failure of any one master should not impact the ability to update resource record information and that no manual steps are required to promote a former primary slave. As in the multi-tiered approach, the DHCP server and other slave servers (if any) would need to be configured with the IP addresses of all masters for each zone. Thus, when a given master DNS server in unreachable, the DHCP and slave DNS servers will automatically attempt to reach another master. This update process is shown in the nominal case in Figure 4.

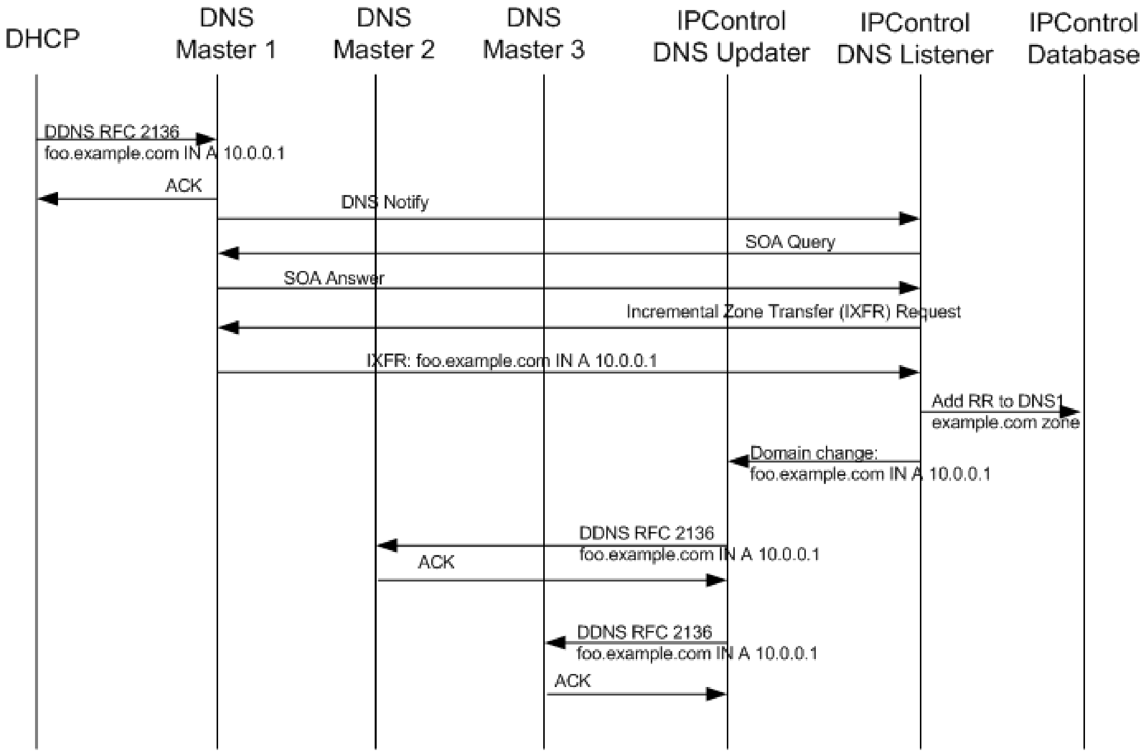


Figure 4: Multi-master update process

In Figure 4, three DNS servers are configured as master for the example.com zone in this case. The multi-master scenario for static zones, i.e., those for which dynamic updates are not supported, entails updating the zone from IPControl and deploying it to all of the masters through the network services tagging feature. This process enables all servers to load the same zone including the same serial number. If your system updates the serial number upon each deployment, you could have a case with the same zone except for the SOA serial number field, which makes synchronization more challenging.

While configuring the same serial number initially eases the zone synchronization process, challenges remain particularly for dynamic zones as different masters receive updates from different sources, e.g., different DHCP servers. For example, if DNS Master server 1 receives an update for foo.example.com IN A 10.0.0.1 as shown in Figure 4, it would attempt to update its peer masters by send each a DNS Update message, though controls must be added to assure only updates from non-masters propagate to other masters to prevent perpetual ricochets. If before DNS Master 1 can communicate this update to its peer masters, DNS Master 2 receives an update which deletes foo.example.com, it will ignore the update since it did not have a record for foo.example.com. Therefore, the delete would not be propagated and any masters that had received the initial foo.example.com addition would retain the record. There are other scenarios where synchronization may be lost but suffice it to say a peer update approach using serial numbers is unreliable.

To improve update reliability, we've proposed a DNS Updater service for IPControl in Figure 4. In this manner, the DNS Updater can track each master's serial number independently, and track each incremental update for each master. This enables it to update the other masters with updates from each master. Thus, a master receiving an update would convey the update only to the DNS Listener service, which would in turn, update the IPControl database and stage the update for propagation to the other masters. Figure 5 illustrates the comparable update process in the event of a failure of DNS Master 1.

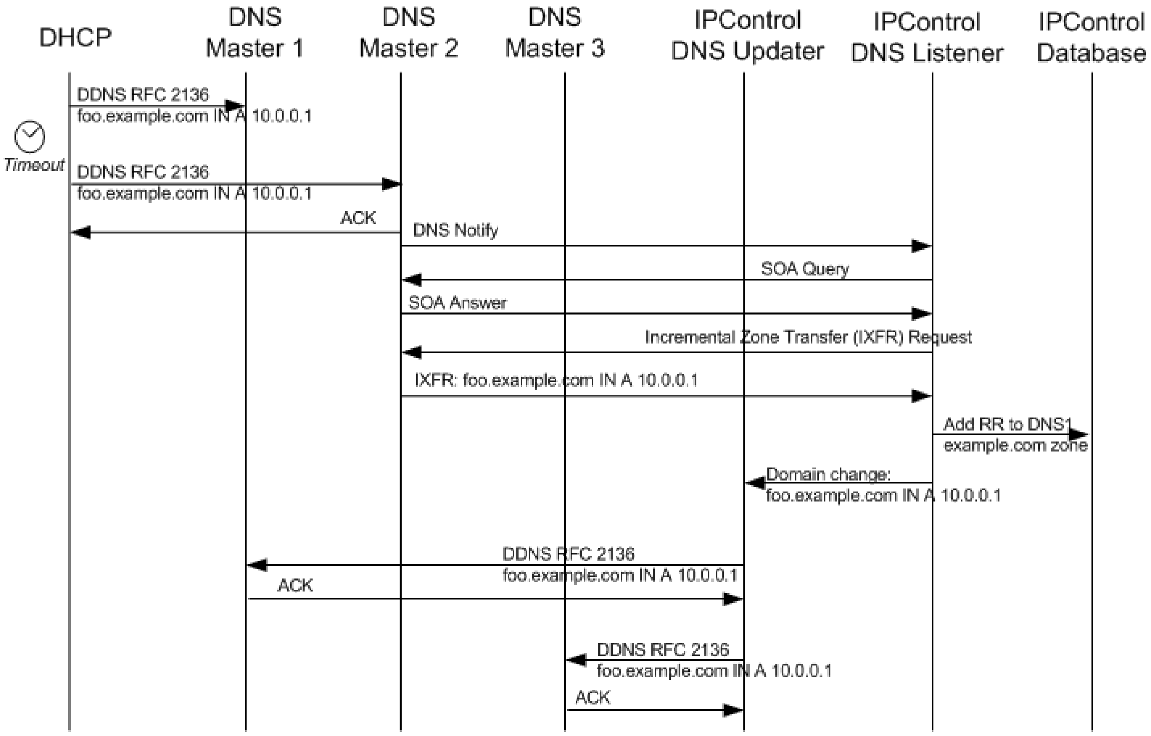


Figure 5: Multi-master with unresponsive master

The process follows the same path as DNS Master 2 is "unaware" of the status of Master 1 and accepts the update from a trusted source as it would any other. It then follows the normal process of updating the DNS Listener and leaving it to the DNS Updater to update its peer masters. The DNS Updater would attempt to update Master 1 and would queue updates until Master 1 returns to service.

This multi-master architecture offers the following advantages:

- Automated operation in the face of the failure of a master DNS server – no manual intervention required.
- Use of Internet protocols for updating multiple DNS servers with updated zone resource record information

This approach however suffers from the following disadvantages:

- There remains a single point of failure with all updates traversing the DNS Listener and DNS Updater services for timely redistribution.
- Updates may take a few minutes to propagate to all authoritative servers during the Notify/IXFR process

Other approaches

Some vendors have attempted to support alternative multi-master architectures such as through the use of LDAP or similar technologies in a grid-like replication scheme with a relatively high degree of marketing success despite their respective inability to offer fool-proof solutions. After all, why would one require a clustered "master" if the distributed database is purportedly bullet-proof? The answer is that you can only send updates to one "master," which obviates the advantage of multi-master over multi-authoritative or multi-tiered approaches.

Unfortunately, despite vendor claims, when you have a "many-to-many-to-many" update architecture as in DNS with many DHCP servers updating any of many DNS servers who in turn update many other DNS servers, each update requires non-zero time, affording the opportunity to lose end-to-end synchronization.

Hardware clustering offers another approach though often at the expense of requiring co-location, which is susceptible to site disasters. Inter-site storage area networks offer another approach for rapid replication but often at the expense of DNS data resolution performance.

Summary

DNS is a critical network service that requires maximum availability for resolving queries and for processing changes to DNS data. We've discussed several approaches to supporting a highly available DNS update architecture in this paper. The following table summarizes the relative benefits and drawbacks of each approach.

Attribute	Architecture			
	Multi-authoritative	Multi-tiered	Multi-master	Grid-like
Internet standard protocols	Yes	Yes	Yes	Yes (for LDAP)
One-to-many updates	Yes	Yes	Yes	Yes
Update propagation time	Fastest (one-layer)	Fast (two-layer)	Fast (two-layer)	Slow (for LDAP)
Failed master resilience	None	Manual effort	Automated	Manual effort
Lossless updates using serial number alone	Yes (single authoritative)	Yes (single authoritative)	No	Yes (single authoritative)
Lossless updates using IPControl	Yes (single authoritative)	Yes (single authoritative)	Yes (via IPControl DNS Updater)	Yes (single authoritative)

All approaches utilize Internet-standard protocols for DNS including DNS Update and DNS Notify. Incremental zone transfers are also supported by the multi-authoritative, multi-tiered and multi-master approaches; for LDAP-based replication architectures, LDAP is not a DNS protocol but it is an Internet standard. Other vendor replication strategies may use proprietary mechanisms.

All approaches likewise support one-to-many updates, featuring the ability to send one DDNS update to a master, which in turn propagates the update to other masters and slaves. In terms of the relative update propagation latency, the multi-authoritative approach is the fastest: the update is sent to the master and it notifies and updates each slave, offering a one-layer update approach. The Multi-tiered and multi-master approach require updating of other masters, then of corresponding slaves. In the case of multi-tiered, this approach offers similar performance as multi-authoritative since the primary master updates the primary slave and other slaves at nearly the same time, while each master in a multi-master architecture may have its own set of slaves to update, creating a two-layered update architecture.

Failed master resilience relates the ability of each solution to respond to an outage of a DNS master. In the multi-authoritative case, a single master exists and its failure inhibits the ability to perform updates. For the multi-tiered case, the failure of the master requires detection by an administrator and a manual promotion of the primary slave to assume the role of master, thereby resuming updatability via the newly promoted master. In the multi-master scenario outlined in this white paper, as long as the DHCP server can automatically send an unacknowledged DNS update message to a master other than the one to which it typically sends them, the master receiving the update would be able to notify the IPControl DNS Listener, which in turn may propagate the update to the other active masters via the IPControl DNS Updater.

In the grid-like case, the same stipulation exists. Some vendor solutions do not permit the automated transmission of an unacknowledged DNS update to a secondary master. In such a case, manual intervention is required to promote another master and reconfigure the DHCP servers accordingly.

Lossless updates refers to the ability of the architecture to capture all updates then unambiguously and in the proper sequencing, apply them to all other masters. Each architecture except multi-master can provide lossless updates based solely on serial numbers. Again, if using a grid-like solution this may be "N/A" is a proprietary update process is applied. As we discussed in the multi-master section, it is conceivable that different masters may have the same serial number value but a different set of corresponding resource records; therefore, resource record loss is possible if not likely. Applying the IPControl DNS Updater solution, one may attain lossless updates thanks to the marshaling of updates via the DNS Updater service.

The objective of providing highly available DNS services with the ability to reliably perform timely DNS data changes can be met under various approaches, though each suffers its particular drawbacks. The multi-authoritative approach suffers from the inability to automatically recover from a failed master; the multi-tiered approach likewise performs no automated recovery though the process for recovery is staged albeit manually. The multi-master case suffers from exposure to lost resource record updates and while the IPControl DNS Updater service eliminates loss, it itself does present a single point of failure though it can be made redundant. The grid-like approach suffers potentially from resource record loss or required manual effort to promote a backup master and/or repoint DHCP servers.

The bottom line is that there is no one approach that can meet the objective with zero tradeoffs. One must assess the tradeoffs and determine which are most bearable with respect to the cost and complexity of implementing a corresponding solution seeking to minimize loss, update latency, or manual processes within a resilient DNS masters architecture.

About BT Diamond IP

BT Diamond IP is a leading provider of software and appliance products and services that help customers effectively manage complex IP networks. Our IP management solutions help businesses more efficiently manage IP address space across mid-to-very large sized enterprise and service provider networks. These products include IPControl™ for comprehensive IP address management and Sapphire hardware and virtual appliances for DNS/DHCP services deployment. Our cable firmware management product, ImageControl™, helps broadband cable operators automate and simplify the process of upgrading and maintaining firmware on DOCSIS devices in the field. Our customers include regional, national and global service providers and enterprises in all major industries.

For more information, please contact us directly at +1-610-321-9000 worldwide, email to btdiamondip-sales@bt.com or consult www.btdiamondip.com.

IPControl is a trademark of BT Americas, Inc.