14

IP ADDRESS MANAGEMENT PRACTICES

In the Preface of this book, we stated that the practice of IP address management (IPAM) entails the application of network management disciplines to IP address space and associated network services. Because IP addresses and associated DHCP and DNS functions are so foundational to IP services and applications running over a network, these functions must be prudently managed, much as other critical network infrastructure elements are managed. It's a small leap to consider DNS and DHCP servers as network elements, as they provide critical IP services to clients on an IP network. While not in-band or on the data path for user IP traffic like traditional network elements, they provide necessary services required to make such in-band data paths possible and usable. From a telephony Intelligent Network analogy, DNS and DHCP are akin to Network Control Points in providing lookup and addressing infromation. So it follows that centralized management of these servers is equally wise and beneficial[*].

[*] Much of the content of this chapter mirrors that of similarly titled Chapter 6 of Ref. 11.

IP Address Management: Principles and Practice, by Timothy Rooney

The most commonly applied network management approach is that of the FCAPS* model for network management. The Information Technology Infrastructure Library, ITIL®, has emerged as a popular set of guidelines for managing enterprise IT infrastructures. Developed by the U.K. Office of Government and Commerce (OGC), ITIL is a best practices framework with the perspective of the IT organization as a service provider to the enterprise. We'll discuss common IP address management tasks within the context of the FCAPS model, then relate functional mapping of these tasks to ITIL process areas toward the end of the chapter.

14.1 FCAPS SUMMARY

The FCAPS model covers the following key functions within the practice of network management:

- *F = Fault Management*. Involves monitoring and detection of network faults with the ability to diagnose, isolate, and resolve them. As network elements such as routers, servers, and switches are monitored to detect faults or outages, DHCP and DNS services should likewise be monitored. Appropriate workaround mechanisms such as providing for high availability services may also be implemented.
- *C = Configuration Management*. Entails accurate configuration and backups of network elements, including DHCP and DNS servers. Accurate and timely configuration of network elements reduces provisioning errors and time intervals within change management windows.
- *A = Accounting Management*. Involves tracking and policing of usage of network resources with respect to business quotas or customer entitlements. Aspects of IP management regarding access control policies, address utilization with respect to business parameters, and monitoring service level agreement (SLA) compliance fall within accounting management.
- *P = Performance Management*. Deals with tracking performance of network elements and services, along with resource utilization. Tracking of IP address utilization and DHCP/DNS server performance are key requirements for effective IP address management.
- *S = Security Management*. Includes the securing of information regarding the network and its users, providing access controls, as well as audit logging and security breach detection. Security management for IP address management includes IP address access policies, auditing, DNS and DHCP security, and rogue or illicit device detection on the network.

* FCAPS is defined in ITU standard M.3400 (192) as part of the Telecommunications Management Network (TMN) framework for managing data networks.

14.2 COMMON IP MANAGEMENT TASKS

Using the basic FCAPS functional categorization, we'll discuss the most common IP management tasks, starting with "configuration," then move on to the other categories. Some functions may likely require the use of multiple management systems depending on your IP management system capabilities. For example, if your IP management system consists of a spreadsheet, you'll need another tool to perform fault management functions. Likewise for commercial IP management systems, varying subsets of functions and tasks will be available natively within the system, while others will require supplemental systems.

14.3 CONFIGURATION MANAGEMENT

When most people think of IPAM, they primarily consider it a configuration management mechanism. Early IPAM systems in fact focused solely on configuration management, though many have expanded into other aspects of FCAPS over time. Nevertheless, configuration management remains a fundamental function of IPAM. In this section, we'll discuss common tasks required when managing IP address space and DHCP/DNS server configurations. These tasks relate to the day-to-day activities of an IP address planner with respect to moves, adds, and changes for IP addresses, subnets, address space, domains, and other aspects of DHCP and DNS configuration.

Configuration management within the context of IPAM entails the configuring of DHCP and DNS servers for lease and parameter assignment and name resolution, respectively. This involves at minimum, configuration of IPAM-related information, that is, address pools and associated parameters and DNS configuration and zone files. The configuration process may also entail configuration of high availablilty deployments and server level configuration parameters, for server-based or appliance-based DHCP/DNS servers.

The result of the configuration management function is that each of the DHCP and DNS servers within the network is configured with its files or parameters necessary to perform its respective role in the network, for example, primary DHCP server for a set of address pools, failover DHCP configuration, DNS zones, parameters, and options. From this perspective, the goal is to base each DHCP and DNS server's configuration on its type (e.g., ISC, Microsoft, etc.), its role in deployment, and the portion of the network it is serving. The portion of the network relates to the association of a set of DNS domains, subnets, and address pools assigned to each server, and should align with the overall IPAM plan for address space and domains.

Configuration of routers with new, moved, or deleted subnets, as well as relay agent information regarding which DHCP servers to which to relay DHCP packets is another function closely tied to IP management. Few IPAM systems on the market perform this level of router integration natively. Historically, the IP or server teams were distinct from router teams, so inter-team automation was discouraged; after all, if a router ended up being misconfigured, it would come back to the router team. Some IPAM systems enable automation of this process nonetheless, natively or via an API call, which can "hook" the

output of an IPAM system subnet allocation to the input of a router configuration tool. The brute force method likely entails sending an email to the router team after a subnet has been allocated in the IP inventory database or spreadsheet.

14.3.1 Address Allocation Tasks

Address Block Allocation. Starting at the top of the IPAM food chain, consider the tasks required to perform top-level block allocations using the processes we discussed in Chapter 3. Allocation of address space hierarchically from the top-down, planning address space allocations must consider business requirements with respect to address capacity for each application and user community at each site from the bottom-up. Ultimately, each site will be served from the respective allocation, so capacity planning should incorporate addressing needs at each current and planned future site.

If you don't have time or resources to conduct a full capacity analysis, one rule of thumb for enterprise organizations is to consider the number of employees at each location and multiply this number by four. This quantity provides a rough estimate and accounts for each employee's devices as well as infrastructure devices like routers and servers. On the other hand, if you have plentiful address space for the size of your organization, you may just want to allocate uniformly, as we illustrated in Chapter 3.

Once address capacity has been quantified per site, consider the routing topology and how to best model the addressing hierarchy. Using a core-regional-access router topology as with IPAM Worldwide lends itself to a corresponding mapping of addressing hierarchy. Such a topology features a backbone or core network feeding regional networks, which in turn feed access or local networks. Routers serve as topological interfaces and provide aggregation of downstream networks. Now integrate the capacity data with the topology to identify the roll-up of address space at each hierarchy level.

Let's illustrate this integration by example. IPAM Worldwide's topology features a core network serving the global continental level. The regional level subdivides the continents of North America and Europe. With 17,000 employees mapping to roughly 75,000 IP addresses, our 10.0.0.0/8 network should provide plenty of capacity with over 16 million IP addresses, let alone our IPv6 space. Thus, IPAM Worldwide's IP planners decide to utilize a *uniform* allocation strategy as much as possible. The largest distribution center is in Norristown with about 450 employees and the largest branch office of Quincy is planning expansion up to 200 employees. Hence, each distribution center will receive a /23 allocation (510 usable IP addresses) per application[*] and each branch office will receive a /24 (254 usable addresses) per application. This provides ample address space for each site with room for growth.

The region with the most offices, North America East, contains 8 distribution centers and 9 branch offices. The corresponding address space rollup per application (8 /23s + 9 /24s) is roughly a /19[†]. To provide adequate address capacity for growth, IPAM

[*] If we were to assign one monolithic block to these sites, we'd likely require a quantity of IP addresses totaling four times the number of employees, but we're allocating multiple like-sized blocks instead.

[†] Eight /23s = one /20, and eight /24s = one /21. One /20 + one /21 + one /24 is over 3/4 of a /19.

Worldwide's IP planner allocates a /18 for each application in each region. After these sizing guidelines have been defined, the execution process detailed in Chapter 3 for IPAM Worldwide may ensue.

A more intensive strategy may alternatively be employed to allocate to each site only what is needed. This *as-needed* approach requires more precise determination and tracking over time of IP address capacity requirements at each site to assure adequate capacity deployment. This approach is more intensive but makes for better utilization of the address space, which may be required for larger organizations or service providers. The same process is used for block allocation, though the math and tracking requirements become a bit more rigorous, especially with a mix of differently sized allocations. The requirement to proactively monitor address utilization increases when allocations are "right-sized" to conserve address space, where extra "fudge factors" in allocations are kept to a minimum.

Based on the selected DHCP and DNS server deployment strategy and needs of the expansion, you should map out server sizing regarding the number of servers of a given size of each type needed and target locations. Based on this plan, procurement of the servers, shipping/receiving, then base level server configuration can ensue. While we haven't added any suballocations yet at this point in the process, this base DHCP/DNS configuration would include basic policy definitions, as well as zones corresponding to the domain plan for the additional address space.

For base and subsequent address block allocations, updating the address plan is a necessary first step. But there's more to be done. To implement the plan, the allocated address space should be configured in the core routers to enable dynamic updating of routing tables. Updating of relay agent information in routers to relay to the DHCP server(s) is another required task, though this is more commonly performed during the subnet allocation task. Additional housekeeping tasks may be necessary to add the newly allocated address space to server access control lists (ACLs) at the network interface level and also at the DNS service level regarding "allow" options such as allow-query, allow-recursion, and so on, as well as view definitions if appropriate.

In summary, the task of address block allocation includes the following subtasks:

- Identify sites requiring IP space and quantities of users or IP devices required per site.
- Determine the routing topology in terms of address aggregation requirements.
- Identify the minimum allocation at each level of the topology considering growth plans, and employ allocation policies such as a uniform or as-needed strategy.
- Identify free address space within the IP address inventory and allocate a block of the selected size.
- Design, procure, install, and configure DHCP and DNS servers as required.
- Update router configurations with the allocated network and relay agent information.
- Update DHCP and DNS ACL configurations if appropriate.

- Manage the overall allocation process to track locations and servers coming on line. Subnet allocations per location are covered next.

Subnet Allocation. After the baseline address allocation has been deployed, the foundation is in place for subnet allocations that support individual IP addresses for routers and hosts. Business initiatives will likely drive subnet allocations: new sites requiring IP addresses due to expansion of the business, plans for new service offerings such as voice over IP, and even a merger or acquisition each can heavily impact the IP address plan. A similar process with respect to sizing up expected capacity requirements, mapping capacity rollups to the supporting routing topology, and consideration of free address capacity and allocation policies can be used for subsequent allocations.

This basic task of subnet allocation involves the determination of a subnet that is available which rolls up with the address allocation plan for the given location and application, and assignment of the subnet in the IP address plan "database." For example, if IPAM Worldwide decides to open a new distribution center in Portland, Oregon, IP planners would access our IP inventory spreadsheet to identify available address space. The address allocation breakdown of North America West data space (10.32.128.0/18) was illustrated in Chapter 3 and is shown below.

N.A. West Data	10.32.128.0/18	***00001010 00100000 10***000000 00000000
San Fran. Site	10.32.128.0/23	***00001010 00100000 1000000***0 00000000
Denver Site	10.32.130.0/23	***00001010 00100000 1000001***0 00000000
Vancouver Site	10.32.132.0/23	***00001010 00100000 1000010***0 00000000
Phoenix Site	10.32.134.0/23	***00001010 00100000 1000011***0 00000000
Calgary Site	10.32.136.0/24	***00001010 00100000 10001000*** 00000000
Albuquerque Site	10.32.137.0/24	***00001010 00100000 10001001*** 00000000
Salt Lake City Site	10.32.138.0/24	***00001010 00100000 10001010*** 00000000
Boulder Site	10.32.139.0/24	***00001010 00100000 10001011*** 00000000
Edmonton Site	10.32.140.0/24	***00001010 00100000 10001100*** 00000000
Sacramento Site	10.32.141.0/24	***00001010 00100000 10001101*** 00000000
Anaheim Site	10.32.142.0/24	***00001010 00100000 10001110*** 00000000
Free Space	10.32.143.0/24	***00001010 00100000 10001111*** 00000000
Free Space	10.32.144.0/20	***00001010 00100000 1001***0000 00000000
Free Space	10.32.160.0/19	***00001010 00100000 101***00000 00000000

As we discussed in Chapter 3, by using a best-fit approach, we should consider the smallest free block for allocation. We see from this table that we have a /24 available, but this is too small for our Portland distribution center, which requires a /23. We'll look to the next smallest block, 10.32.144.0/20 and allocate our /23 from this block. We thus allocate 10.32.144.0/23 to Portland, and the remainder of the original /20 block consists

of 10.32.152.0/21, 10.32.148.0/22, and 10.32.146.0/23 as shown in the resultant table below.

N.A. West Data	10.32.128.0/18	***00001010 00100000 10***000000 00000000
San Fran. Site	10.32.128.0/23	***00001010 00100000 1000000***0 00000000
Denver Site	10.32.130.0/23	***00001010 00100000 1000001***0 00000000
Vancouver Site	10.32.132.0/23	***00001010 00100000 1000010***0 00000000
Phoenix Site	10.32.134.0/23	***00001010 00100000 1000011***0 00000000
Calgary Site	10.32.136.0/24	***00001010 00100000 10001000*** 00000000
Albuquerque Site	10.32.137.0/24	***00001010 00100000 10001001*** 00000000
Salt Lake City Site	10.32.138.0/24	***00001010 00100000 10001010*** 00000000
Boulder Site	10.32.139.0/24	***00001010 00100000 10001011*** 00000000
Edmonton Site	10.32.140.0/24	***00001010 00100000 10001100*** 00000000
Sacramento Site	10.32.141.0/24	***00001010 00100000 10001101*** 00000000
Anaheim Site	10.32.142.0/24	***00001010 00100000 10001110*** 00000000
Portland Site	10.32.144.0/23	***00001010 00100000 1001000***0 00000000
Free Space	10.32.143.0/24	***00001010 00100000 10001111*** 00000000
Free Space	10.32.146.0/23	***00001010 00100000 1001001***0 00000000
Free Space	10.32.148.0/22	***00001010 00100000 100101***00 00000000
Free Space	10.32.152.0/21	***00001010 00100000 10011***000 00000000
Free Space	10.32.160.0/19	***00001010 00100000 101***00000 00000000

Note that our free 10.32.143.0/24 block is now enclosed or surrounded by assigned blocks. A future requirement for a /24 or smaller block can use this space, but otherwise it is unusable. Applying a best-fit approach seeks to minimize these "orphaned" blocks, though as we see, they still may be rendered.

In addition to identifying and recording the allocated subnet, the subnet allocation process requires provisioning of the subnet address on the appropriate router interface. Some individual IP addresses on the subnet need to be assigned to infrastructure devices like routers and servers. Defining and updating DHCP server configurations is also required to account for address pool(s) and corresponding DHCP options and/or client class parameters needed for devices that will require DHCP on the allocated subnet.

Devices to be assigned addresses on the subnet now and in the future will likely require name resolution information in DNS. At a minimum, this information applies to a forward domain for domain name-to-IP address lookup and a reverse domain for the IP address-to-name lookup. This requires defining and updating DNS server configurations with domain updates (e.g., in-addr.arpa and ip6.arpa domain(s)) and resource record updates for name servers and statically assigned addresses. Of course, these domains must exist or must be provisioned and configured on the respective DNS server.

Depending on your domain topology, adding a new subnet to a location may utilize an existing domain, though this is not necessarily the case. A new domain may need to be

defined and configured as a subdomain or as a new zone on the appropriate DNS servers. In the same way, the reverse domain corresponding to the subnet address may need to be added as well, unless a higher layer in-addr.arpa or ip6.arpa zone will host the corresponding PTR resource records.

The subnet allocation process illustrates the tight interrelationship among address allocation, assignment, and DHCP and DNS server configuration tasks. Depending on your business processes, subnets may be allocated or reserved prior to address assignment and DHCP/DNS configuration. Nonetheless, this complete set of steps will typically be required to bring a subnet into production:

- Identify free address space within the scope of the topology where the subnet is needed.
- Allocate a subnet of the required size from the appropriate address space and record the allocation in the IP address plan.
- Update router configurations regarding the allocated network.
- Assign and provision manually assigned addresses for routers, servers or other subnet infrastructure devices.
- Design and configure DHCP address pools if necessary to serve dynamic hosts on the subnet. This may require association of options, directives and client classes based on requirements of devices planned for use of the address pool(s).
- Define new DNS domains required to serve hosts on the subnet, define resource records for infrastructure or static devices within new or existing domains and configure appropriate DNS servers.*
- Complete the allocation process by confirming provisioning and reachability of the subnet, as well as by verifying corresponding DHCP and DNS configurations.

IP Address Assignment. Assigning, deassigning, and reassigning IP addresses to individual hosts is usually the most frequent IP management activity in most organizations. This is typically associated with deployment, redeployment, or decommissioning of devices, including routers, servers, printers, and the like. In terms of address assignment, the IP address inventory database must be consulted to identify an available IP address. If possible, it would be useful to ping the IP address to be assigned just to verify accuracy of the inventory, though we'll discuss the process of overall inventory assurance as a separate task. The IP address to be assigned should then be denoted as assigned to the given device in the inventory database.

The actual physical IP address assignment may be performed by manually (statically) configuring the device, by autoconfiguration, or by using DHCP (in this case, we'll assume Manual DHCP is used to assign the designated IP address to the corresponding host). In the static assignment case, the assigned address must be configured directly on the device,

* In some networks, pre-seeding of resource records for DHCP addresses is required to permit users of these addresses to appear in DNS (e.g., to facilitate VPN connections, which require the presence of a PTR record) without performing dynamic updates.

so unless the IP address assigner is also responsible for the physical assignment, this process would entail an email or phone call to the device owner conveying the assigned IP address information to be entered. With autoconfiguration, this bottom-up assignment process is more of a detection issue than top-down assignment. When using Manual DHCP, an entry in the appropriate DHCP server(s) configuration file would be necessary to map the device's hardware address to the assigned IP address.

Most devices with IP addresses will require corresponding DNS resource records to enable reachability by name. Using the DHCP method of address assignment, the DHCP server can be configured to update a master DNS server upon assignment of the IP address. This update would affect the forward domain for domain name-to-IP address (A/AAAA) lookup and the reverse domain for the reverse (PTR) lookup. A similar DNS update task would be required if assigning the address manually. Updating DNS with this new host information may entail editing or updating the corresponding zone files on the server or by sending dynamic updates.

You may not want an autoconfigured device to update DNS on its own, at least on an enterprise network, though this may be suitable for a community or ad hoc network. Identifying the presence of a newly autoconfigured device to manually update DNS presents its own challenge! If such devices require resolution information in DNS, use of a router log or subnet snooping utility may be necessary to identify the IPv6 address.

In summary, the task of IP address assignment includes the following subtasks:

- Determine how the device will obtain its IP address: via manual configuration, autoconfiguration, or via DHCP.
  - If Dynamic DHCP or Automatic DHCP, determine if current address pools, if any, on the subnet have capacity to support the device; if so, this task completes; if not, configure an address pool of the corresponding DHCP type on the DHCP server along with necessary option parameters.
  - If Manual DHCP, identify a free IP address within the subnet where the device is located and assign the address to the device by configuring the DHCP server to reserve or assign a Manual DHCP address for the device's MAC address.
  - If manually configured on the device, identify a free IP address within the subnet where the device is located and assign the address to the device. Have the assigned static IP address configured on the device manually.
  - In all cases, update the IP address plan with the assigned address, whether a spreadsheet or other IPAM tool.
- Determine if DNS resource records need to be manually created and updated. This would generally be the case for statically assigned addresses. For DHCP-assigned devices, the DHCP server can be configured to perform dynamic updates, though in some cases where dynamic updates are not feasible or allowed by policy, manual updating of corresponding resource records may be required.
- Verify completion of the address assignment process by pinging the address successfully and verifying its resource records in DNS. For devices assigned an

address via an address pool, verification may not be needed; however, if it is, the address may not be known *a priori*. Locating the device's MAC address in the DHCP lease file, followed by a ping of the corresponding address confirms its assignment in this case.

14.3.2 Address Deletion Tasks

As we've illustrated, address allocation is a top-down process, with allocation of hierarchical blocks, from which subnets can be allocated, from with IP addresses can be assigned. Deletion of address space requires the inverse operation and is necessarily bottom-up. Deleting an address block before the underlying blocks, subnets, and IP addresses have been deleted would strand these underlying elements, so unless you enjoy mass chaos, a more controlled process is warranted.

Deleting IP Addresses. Deleting an IP address is relatively straightforward: delete or free up the IP address in the IP inventory, removing the M-DHCP entry from DHCP if appropriate releasing the lease, and removing associated DNS resource records. However, care must be taken to assure the address has been relinquished by the device and that DHCP and DNS updates have been completed before assigning the address to another device. For example, simply deleting a lease on a DHCP server does not force the client holding that lease to relinquish it. The DHCP Force-Renew message was designed to force a DHCP client to enter the Renewing state to enable a server to potentially NAK the client's attempt to renew the lease, thereby freeing the address. However, Force-Renew has not been widely implemented.

Denoting the address as in a state of "pending deletion" or something similar would alert other administrators not to assign that address to another device until confirmation is received of its availability. This confirmation process entails pinging the address, perhaps successively over several days, and confirming the deletion of its associated data in DNS and DHCP servers.

Deleting Subnets. Deleting a subnet may be required when closing a site or consolidating address space. Devices with IP addresses on the subnet to be deleted should be moved or decommissioned such that the subnet is free of address assignments (other than perhaps subnet-serving routers). After all IP addresses have been verified as free, the subnet may be reclaimed into the free address space for future allocation.

Upon freeing up of a subnet, it may be possible to join the freed space with a contiguous free address block, creating a larger free block. Following our IPAM Worldwide North America West Data block example from the Subnet Allocation section above, if IPAM Worldwide decides to close the Anaheim branch office, its address space, 10.32.142.0/24, now considered free, is contiguous with the 10.32.143.0/24 block. We could join these two blocks into a single free block, 10.32.142.0/23. Doing so now makes it clear that this /23 could be assigned to a future distribution center for example.

Deleting Blocks. Address block deletion may result from the withdrawal from a major business market or consolidation of sites, among other reasons. Generally all

downstream IP addresses, subnets, address pools, resource records, and domains should first be decommissioned before the macro level block can be freed up for future assignment consideration. Thus, after the individual delete IP address tasks and delete subnet tasks within the target block have been completed, the block itself may be freed up. As with a modest to large allocation task, project planning resources may be required to verify deletions up the hierarchy. As in the delete subnet task, freed block space may be joined with contiguous free space. As with block allocations, additional housekeeping tasks related to DHCP and DNS ACL configurations should be considered with respect to address pools, domains, ACLs, and resource records.

14.3.3 Address Renumbering or Movement Tasks

Moving, or renumbering of address blocks, subnets or individual addresses combines the allocation process with the deletion process. The allocation process, as described above, should be performed from a top-down perspective to allocate space to which underlying subnets and IP addresses will be moved. The deletion process frees up address space from the bottom-up as addresses are moved to the target allocated space. In essence, the size of the scope of the addresses to be moved must be allocated to accommodate the addresses to be moved, temporarily doubling the address space associated with this set of devices. As addresses are moved, the former address space can be freed up, returning address allocations to previous levels.

IP Address Moves. Moving an IP address can be considered a combination of assigning an IP address on the destination subnet and deleting the IP address on the current subnet. Depending on the method of address assignment and the type of move, different tactics can be used. The type of move relates to physical movement of a device to a different subnet (physical move) versus the reassignment of the IP address on the same or a different subnet (logical move). A physical move of a nonmobile IP device will typically involve a "reboot" of each IP device, which affords more control of the address assignment process.

PHYSICAL MOVES. Physical moves imply powering down, moving, then powering up devices at the destination location. For Dynamic DHCP and Automatic DHCP assigned devices, if an entire pool is being moved, the destination pool should be setup on a [same or different] DHCP server. Make sure the router(s) serving the destination subnet are configured to relay DHCP packets to the DHCP server configured with the new pool. When these devices power up, they will likely attempt to renew the most recent lease they possessed on the old subnet. Make sure any Automatic DHCP devices issue DHCPREQUESTs upon power up and don't just continue using their old IP lease; if they assume the old [infinite] is valid, manual intervention will be required to reset the device's address. Otherwise, the DHCP server will NAK the DHCPREQUEST attempt by each client. Clients will revert to the Init-Reboot state and issue a DHCPDISCOVER packet to obtain a new lease. The DHCP server obliges with a lease within the new destination pool. A similar process may be used for DHCPv6 clients. Once all devices have physically moved, the pool serving the old subnet may be decommissioned.

Physical movement of a M-DHCP device entails creating the M-DHCP entry in the DHCP server serving the new subnet and deleting the entry on the former DHCP server. If the same DHCP server is being used, simply edit the IP address associated with the device's MAC address. When the device powers up on the new subnet, it should follow a similar process to Dynamic DHCP and Automatic DHCP, with a DHCPREQUEST attempt, which the DHCP server NAKs, followed by reversion to issuing a DHCPDISCOVER and address reassignment using the standard DHCP process.

Moving a device that autoconfigures its IPv6 address will lead to the device detecting its new subnet via router discovery along with corresponding subnet policies including the availability of DHCPv6 services. If autoconfiguring its address, the device autoconfigures then verifies its uniqueness through duplicate address detection. If using DHCPv6, the normal DHCPv6 process is followed to obtain an IPv6 address and associated parameters. In some cases (i.e., when the O bit is set in the router advertisement), both autoconfiguration and DHCPv6 may be used.

Updating of DNS resource records may be performed by the DHCP server or manually if dynamic updates are prohibited for these DHCP cases.

Physical moves of manually configured devices requires assignment of an address from the IP inventory, and manually configuring the new IP address in the device as it powers up on the new subnet. At this point, the old address can be freed up, though an interim "pending delete" state may be useful in preventing premature reassignment of the corresponding address prior to verification of address availability. DNS resource records should be updated as well to reflect the device's new IP address.

In all of these cases, the IP inventory should be utilized to identify free address(es) on the destination subnet or pool, and to free up addresses on the old subnet as well as corresponding DNS resource records as device moves are confirmed.

Logical Moves. Logical moves are a bit more challenging as they do not necessarily involve a device reinitializing. For DHCP devices, an address pool containing the destination IP addresses should be configured on the [same or different] DHCP server. The lease time for the pool or device should be stepped down in advance of the move date. For example, if a normal lease time is 1 week, it should be lowered to 1 day for example during the week leading up to the move and to 2–6 h on the day of the move. A device may have renewed a weeklong lease just before you changed the lease time to days, so it will not attempt to renew until halfway through the week (or based on your T1 time option setting). Thus if your nominal lease time is 2 weeks, ratchet down the lease time 2 weeks before the planned move. On the day of the move, set the lease time to a minimum* time if it's important that all devices move at nearly the same time. If move coincidence is not critical, leaving lease times on the order of hours should yield a complete move within a few hours.

In this scenario, it's recommended that the DHCP server perform DNS updates if possible to more closely map DNS information updates with address changes. Manual

* Minimum time can be on the order of minutes or hours depending on network traffic and server performance considerations. The shorter the lease time, the more DHCP packets will be sent but the more time-aligned the move of DHCP clients can be orchestrated.

intervention of A-DHCP devices may be necessary unless they do adhere to lease renewal policies despite possessing infinite leases.

Movement of manually addressed devices follows the same process as in physical movement. A destination IP address is assigned from the IP inventory, and the new IP address is configured on the device. Once confirmed, the old address can be freed up. DNS resource records should be updated as well to reflect the device's new IP address.

Logical movement of an autoconfigured device can be performed by configuring the router serving the corresponding subnet to ratchet down the preferred and valid address lifetime values it advertises during the neighbor (router) discovery process. Shortening these timer values for the address prefix from which the device is being moved while introducing the new prefix with a "normal" address lifetimes will enable autoconfigured devices to perform this logical move automatically. Once all devices have moved and the valid lifetime of the former prefix expires, the prefix can be removed.

Subnet Moves. Moving a subnet could involve one of two results: movement of the subnet and its assigned IP addresses to another router interface, preserving the current address assignment or movement to another router interface, requiring a new subnet address. We'll include the subnet renumbering task with the latter case as it too results in a new subnet address though without necessarily moving the subnet to another router interface. The first case requires consideration of address space rollup within the hierarchy but generally consists of modifying and verifying router provisioning compliance with the plan, as well as updates to routing tables and DHCP Relay addresses as necessary.

Movement of a subnet due to a physical move or a higher level renumbering requires a bit more work. A physical movement where devices are physically moved, for example, when an office is moved, is inherently disruptive. The destination subnet may be allocated and provisioned on the destination router interface, along with the other tasks described above related to reserving static addresses and updating DHCP and DNS configurations. When each moved device plugs in, it will need to be manually readdressed with the new address and/or obtain a DHCP lease on a pool relevant to the subnet as described above for IP address moves. Logical subnet moves or renumbering likewise follows the logical IP address move process for each device.

After all devices have been moved from the old subnet to the new, the old subnet may be freed up following the delete subnet process.

Block Moves. Moving macro level blocks with underlying subnets and IP addresses requires careful project planning and execution. The allocation of the destination block should follow those tasks outlined for block allocation. Assuming a move is for renumbering only, a like-sized destination block should be allocated. If the move is motivated by or otherwise spurs the opportunity for address consolidation or expansion, the destination allocation should be sized based on underlying capacity requirements and topology architecture as discussed in the Block Allocation section. Once the allocation has been made, sub-allocations and subnet allocations may begin. IP

addresses and pools can then be moved following the process described for IP address moves. As IP addresses and subnets completely move from their old assignments, these can be decommissioned or freed up when moves have been confirmed and their corresponding resource records removed.

14.3.4 Block/Subnet Splits

Splitting an address block entails the creation of two or more smaller sized blocks from a given source block. Splits may be necessary to free up address space or even as a means of suballocation of address space. In the former case, the addresses within a subnet may be consolidated to the first half of the subnet, freeing up assignments in the second half. In this scenario splitting the block yields an occupied subnet (first half) and a free subnet (second half). Some organizations have historically allocated regional blocks, then split them to assign sub-blocks and subnets lower in the address hierarchy. In some sense this is a form of block allocation.

Note that splitting a block in two will render two formerly usable addresses generally unusable as the former single network with one network and one broadcast address now has two of each. For example, the 192.168.24.0/24 network has network address 192.168.24.0 and broadcast address 192.168.24.255. Splitting this block into two /25s, 192.168.24.0/25 and 192.168.24.128/25, renders formerly usable address 192.168.24.127 as the new first network's broadcast address and 192.168.24.128 as the second network's network address.

Be cognizant of DNS reverse zone impacts when splitting blocks. If DNS administrative authority for the two resulting subnets remains consolidated under one set of administrators, the original in-addr.arpa or ip6.arpa zone probably does not require modification. However, if a resulting split block or subnet will have its devices administered in DNS by a separate delegated authority, then the original reverse domain will require splitting as well. This entails creation of two reverse zones corresponding to the resulting split subnets and notification to the parent reverse zone administrator of the split in responsibility to properly delegate down the reverse zone tree to the proper set of DNS servers for authoritative information.

Splitting a block need not be restricted to only splitting in half, say a /24 into two /25s. A split may be used to carve out a /23 from a /20, as we did when assigning address space to our new Portland distribution center in the section regarding subnet allocation. This split yielded a /23, which we assigned to Portland, and free space consisting of a /23, a /22, and a /21. In this example, we preserved large blocks following our optimal allocation strategy. Alternatively, we could have simply split our /20 into eight /23s, though this may be wasteful unless same-sized allocations are used by policy, which is the uniform allocation policy, as opposed to the as-needed allocation strategy.

In summary, the process of splitting a block is similar to that of allocating a block. The block to be split is successively divided until the desired block size is attained. Remaining free blocks are either retained or also split to the same size as the desired block to render a uniform block split. DNS implications on the reverse zone tree and

administrative delegation must be considered. And keep in mind that each network resulting from the split results in an additional network and broadcast address.

14.3.5 Block/Subnet Joins

A join combines two contiguous same-sized address blocks or subnets into a single block or subnet. We saw an example of joining blocks in the block deletion section. After freeing up the Anaheim block, 10.32.142.0/24, we joined it to a contiguous free block, 10.32.143.0/24 to create a single 10.32.142.0/23 block. Successive joins may be performed to consolidate smaller chunks of contiguous address space. Joins are only valid for contiguous blocks of the same size. Joining a /25 and a /24 is not valid as the "other /25" not included in the join must remain uniquely identified. However, two contiguous /25s and a neighboring /24 can be joined to form a /23. The two /25s would be joined first to form a /24; then this and the other /24 can be joined to create a /23.

Rolling up of joined blocks may also require an updating of DNS reverse zones to consolidate underlying device resource records into a "joined" reverse zone reflecting the resulting consolidated subnet.

14.3.6 DHCP Server Configuration

DHCP server configuration is a key IPAM task as we covered in Part II of this book. As we've discussed, the address management tasks covered so far have major impact on DHCP server configurations. DHCP server configuration goes beyond address pool creation, movement, and deletion, though the extent of additional functions is constrained by the capabilities of the DHCP server vendor. Key among DHCP server configuration parameters are

- *DHCP Address Pools*. Address ranges and associated DHCP options and server policies for Dynamic, Automatic, and Manual DHCP clients.
- *Client Classes*. Parameter match values (e.g., vendor-class-identifier = "Avaya 4600") and associated allow/deny pools and DHCP options and server policies.
- High availability parameter settings for primary/failover or split scopes.
- Configuration of server activities such as dynamic DNS updates and other server directives and parameters.

The actual server configuration syntax and interface will depend on the server type. For example, ISC DHCP servers can be configured by editing the dhcp.conf file while Microsoft DHCP can be updated using a Windows MMC interface. Both of these and other DHCP vendors also provide command line interfaces or APIs to perform configuration updates. DHCP deployment, covered in Chapter 7, also plays a role in each server's configuration. For these and other products, please consult your vendor's documentation.

Address Assignment/DHCP and IP Address Management. Individual static IP address assignments need to be recorded to assure uniqueness. Within allocated

subnets, DHCP address pools should be tracked to provide an overall view of address assignments within the subnet, whether statically or dynamically assigned. While tracking of individual DHCP leases within a spreadsheet is not readily performed, allocation of address pools within the spreadsheet or database should be performed at the least. This will help assure unique address assignments over time.

This consolidated address assignment data store provides the known level of IP address inventory. The IPAM Worldwide team has assigned a consistent set of IP addresses on each subnet for static devices such as routers, switches and servers. The team has also defined a number of Manual DHCP addresses for printers and address pools for sharing among DHCP client devices like laptops and VoIP phones. We've created a new tab for each site (this spreadsheet is getting quite large!) to inventory individual address as well as address pool assignments. Additional "comment" information is useful to track as well for certain devices, such as vendor contact, support information, asset information, and the like.

In addition to tracking IP address assignments, configuration of the corresponding DHCP server(s) must be performed to enable DHCP clients to obtain addresses. Configuration of the DHCP server entails configuring it with the address ranges corresponding to those assigned within the address plan. In Figure 14.1, we've allocated addresses 10.16.128.50-10.16.129.240 as a DHCP pool, so this range must be defined on a DHCP server. In addition, client class information, options, and other configuration parameters need to be configured on the DHCP server to properly configure different types of clients. This configuration operation may be performed using a text editor for ISC's DHCP server, using Microsoft Management Console (MMC) for Microsoft DHCP servers or using an IPAM tool that supports automated DHCP server configuration for DHCP server types deployed in your network. The main advantages of using an IPAM tool are that IP inventory information readily enables definition of DHCP pools and much of the DHCP server configuration information can be defined in the IPAM system and then applied across multiple DHCP servers, instead of defining this iteratively on multiple servers.

14.3.7 DNS Server Configuration

Like DHCP, DNS server configuration is a critical IPAM function as per Part III of this book. DNS configuration is tightly linked with address allocation, assignment, moves, and deletions as we've seen. These tasks discussed previously affect DNS domains, resource records, and possibly server configuration parameters. Key among DNS server configuration parameters are

- *Domains.* Adding, modifying, or deleting domains/zones on DNS servers.
- *Resource Records.* Adding, modifying, or deleting resource records.
- *Server, View, and Zone Configurations.* Setting and modifying option parameters affecting ACLs, server configuration, and so on.

The actual DNS server configuration syntax will depend on the server type. ISC BIND servers can be configured by editing the named.conf and associated zone files on the

Location	Address Block/Subnet	IP Addresses	Address and Device Type	Comments
San Francisco	10.16.128.0/23	10.16.128.1	Static—Router	SanFran VoIP subnet router 1
		10.16.128.2	Static—Router	SanFran VoIP subnet router 2
		10.16.128.3	Static—Router	SanFran VoIP subnet router HSRP address
		10.16.128.4	Static—DNS Server	Contact Fred Jones for support
		10.16.128.5	Static—FTP Server	
		10.16.128.6	Static—File Server	San Fran secondary
		10.16.128.7	Static—File Server	Backup for Seattle
		10.16.128.8	Static—File Server	Backup for Phoenix
		10.16.128.9		Save for growth
		10.16.128.10	Static—IPPBX	IP PBX-SF1
		10.16.128.11	Static—IPPBX	IP PBX-SF2
		...		
		10.16.128.20	Engineering Lab Server	Contact Engineering for assistance
		10.16.128.21	Engineering Lab Server	Contact Engineering for assistance
		10.16.128.22	Engineering Lab Server	Contact Engineering for assistance
		...		
		10.16.128.50-10.16.129.240	VoIP DHCP Pool	Contact Mary Smith for support
		...		

Figure 14.1. Sample inventory table for IP addresses.

server. DNS servers that support DDNS may also support resource record updates in this manner. The use of `nsupdate` or similar DDNS mechanism provides a means to perform incremental updates without having to manually edit zone text files and reload respective zones, for example, using rndc. DDNS updates apply to resource record adds/changes/deletes only, so any zone or server configuration parameter changes or zone additions or deletions would still require text file editing and reloading of named.conf and/or affected zones. The deployment model for your DNS servers also plays a role in server configuration as we discussed in Chapter 11.

DNS and IP Address Management. Given the direct relationship between IP addresses and reverse domains, hostnames and other host information, it's clear that DNS is a key component of IP address management. Hosts on an IP network are assigned hostnames to facilitate human comprehension and IP addresses to enable communications via IP packets. DNS provides the critical linkage between hostnames and IP addresses, making IP applications easier to use.

From an IP address management perspective, clearly reverse DNS domains have a direct association with IP address block and subnet allocations. These domains are derived directly from their corresponding IP addresses. IPAM Worldwide has secured the ipamworldwide.com domain name from its ISP or domain registry. In so doing, IPAM Worldwide supplied three DNS server addresses to which iterative queries seeking resolution for ipamworldwide.com suffixes can be directed. Assigning web, email, and related Internet-facing servers, this domain suffix can help IPAM Worldwide create a global Internet presence.

Within the organization, this domain name is also used on the intranet. Subdomains are to be defined for the Corporate, Sales, Engineering, and Logistics team. The Engineering subdomain (eng.ipamworldwide.com) has been delegated to Engineering team DNS administrators, while the remaining subdomains will be centrally administered within the IT group. The Engineering team may further create subdomains below eng.ipamworldwide.com without impacting the IT team's administration effort. By delegating the eng subdomain, the IT team is empowering the Engineering team to manage the resolution of all eng subdomain hosts as well as its subdomains.

In keeping with the philosophy of centralizing IP address inventory, it follows that tracking hostnames and resource records associated with each IP address should be performed. Building on our IP inventory spreadsheet we just reviewed for IPAM Worldwide's San Francisco office, we can track this information for individual devices by simply inserting an FQDN column in our spreadsheet as shown in Figure 14.2.

In the above example, we're tracking only the FQDN for each statically defined host. Hosts obtaining leases from the DHCP address pool can have their hostname information updated in DNS via Dynamic DNS. We need to assure that we properly transcribe this inventory information into the DNS server configurations. From this "database," we can derive the A, AAAA, and PTR records corresponding to each host. We could expand the columns on the spreadsheet to track additional resource records associated with given hosts such as CNAME, MX, and so on.

14.3.8 Server Upgrades Management

New versions of DHCP and DNS server software are published periodically to address security vulnerabilities, provide bug fixes, or offer new features. The urgency to perform an upgrade is usually dictated by what's being addressed, with security vulnerabilities certainly being of highest urgency. The upgrade process is typically vendor specific and may require alignment of which hardware platforms and operating systems the upgraded version has been certified to run on. Hopefully the underlying operating system requirements would change only for new feature introductions and not security or bug fixes, but this is governed by vendor policy.

Most vendor DHCP/DNS appliance upgrades roll in the operating system upgrades as necessary within an overall upgrade package. Because the appliance vendor typically provides the operating system with the hardware platform, they should publish

Address Block/Subnet	IP Addresses	Address Type	FQDN	Comments
10.16.128.0/23	10.16.128.1	Static—Router	router-sf01.ipamworldwide.com.	SanFran VoIP subnet router 1
	10.16.128.2	Static—Router	router-sf10.ipamworldwide.com.	SanFran VoIP subnet router 2
	10.16.128.3	Static—Router	router-sf11.ipamworldwide.com.	SanFran VoIP subnet router HSRP address
	10.16.128.4	Static—DNS Server	ns-sf01.ipamworldwide.com.	Contact Fred Jones for support
	10.16.128.5	Static—FTP Server	ftp-sf.ipamworldwide.com.	
	10.16.128.6	Static—File Server	filecab-sf.ipamworldwide.com.	San Fran secondary
	10.16.128.7	Static—File Server	file-dr.ipamworldwide.com.	Backup for Seattle
	10.16.128.8	Static—File Server	file-phx.ipamworldwide.com.	Backup for Phoenix
	10.16.128.9			Save for growth

Figure 14.2. Sample inventory table with FQDNs.

	10.16.128.10	Static—IP PBX	denalo1.corp.ipamworldwide.com.	IP PBX-SF1
	10.16.128.11	Static—IP PBX	denalo2.corp.ipamworldwide.com.	IP PBX-SF2
	...			
	10.16.128.20	Engineering Lab Server	eng-sf1.eng.ipamworldwide.com.	Contact Engineering for assistance
	10.16.128.21	Engineering Lab Server	eng-sf2.eng.ipamworldwide.com.	Contact Engineering for assistance
	10.16.128.22	Engineering Lab Server	eng-sf3.eng.ipamworldwide.com.	Contact Engineering for assistance
	...			
	10.16.128.50-10.16.129.240	DHCP Pool for VoIP Phones		Contact Mary Smith for support
	...			

Figure 14.2. (*continued*)

compatibility upgrades as necessary for newer versions of their DHCP and DNS services. Most appliance solutions enable centralized staging of upgrade packages, with deployment to distributed appliances, vastly simplifying the upgrade process over a software-based upgrade process.

If you're running ISC, Microsoft, or other vendor DHCP or DNS daemons on your own hardware, you'll need to keep apprised of not only DHCP/DNS security updates but also those affecting the corresponding operating system running on the hardware.

14.4 FAULT MANAGEMENT

Fault management encompasses not only fault detection, but alert notification, trouble isolation capabilities, trouble tracking, and problem resolution processes. Monitoring of DHCP and DNS servers for faults and events enables a proactive means of minimizing services outages. In a well-designed network services architecture, clients should be able to obtain leases and resolve domain names despite an

individual DHCP or DNS server outage. Nevertheless, detection of such an outage is important as the outage reduces the number of servers that clients may use to obtain these services thereby raising the vulnerability to service outage in the event of an additional server failure. For example, in a DHCP failover deployment, failure of one server will leave just one server available to service DHCP clients. In such a scenario, detection of the failed server facilitates timely, though not panicked, resolution of the server outage.

14.4.1 Fault Detection

Fault detection may be performed using a variety of methods depending on the capabilities supported by deployed DHCP and DNS servers. These range from proprietary polling or notification, to syslog scanning and/or forwarding, to SNMP polling and trap detection by SNMP-based network management systems. In addition, some commercial IP management systems offer intrasystem or proprietary monitoring, particularly for appliance-based products. Since appliances are fully self-contained solutions, incorporating not only DHCP and DNS services but a hardware platform and operating system, the vendor has the ability to fully access fault information related to the appliance at the hardware, operating system, and DHCP/DNS levels.

In addition to monitoring the state of DHCP and DNS servers, as reported by the servers, it's a good idea to monitor for hung services. This may occur if a service is running but is in a state where it is unable to perform its role in providing leases or resolving DNS queries. This can be detected by analyzing successive polls for lease or query trasactions received and processed, and verifying differential counts greater than zero, assuming normal transaction rates at that particular time of day are nonzero.

An alternative, on-demand form of service testing involves sending the server a DNS query or DHCPDISCOVER (or SOLICIT) packet and verifying receipt of a proper response. This tactic provides some assurance that the services are not only running, but are responding to clients. The bottom line is that some form of service functional fault detection can provide a truer mapping to what an end user may consider a fault or outage.

In addition to monitoring DHCP and DNS servers, monitoring of the IP management system itself provides benefits of assuring access to IP address and DHCP and DNS server configuration information that may otherwise be prevented by an outage. At a minimum, backup or distribution of the data store provides a snapshot to reconstruct the information in the event of a site outage or disaster.

Monitoring of networking equipment and communications links is a common practice for general network monitoring and can provide insights to outages affecting the ability of clients to reach DHCP or DNS servers. This added information can be very helpful in troubeshooting a particular problem or fault.

Fault correlation is the analysis of individual faults received from multiple network elements or management systems to help isolate the root cause of a set of faults. For example, faults from a layer 2 switch, a router, and a WAN access device can be analyzed collectively to suggest that these three faults are related and the likely root cause is a link outage. Fault correlation is a common feature of large scale network

management systems and if your IP management system provides alarm feeds, it may be able to feed into a higher level alert correlation function. Whether fault correlation is performed automatically by a network management system or manually by comparing information from multiple systems, this process exposes a broader set of data for fault analysis with the goal of isolating a fault to a given server, link, or network element.

Fault management capability is an important consideration for those responsible for managing an IP network, and critical DHCP and DNS network services should be among those elements monitored. Mitigation of the impacts of faults may be achieved through deployment of highly available configurations to minimize end user impacts of an outage of any individual component.

14.4.2 Troubleshooting and Fault Resolution

IP Address Troubleshooting. A variety of tools are available to troubleshoot IP assignment, DNS and DHCP faults, some of which may even be provided by your IPAM vendor. To verify or identify IP address assignments, intentional or otherwise, a variety of discovery techniques will prove beneficial. Ranging from a simple ICMP Echo request, ping, traceroute, nmap, or SNMP, a variety of tools may be used to attempt to contact individual hosts or view router or switch ARP tables. Many IPAM systems incorporate at least one form of discovery to provide verification of IP address assignments or to assist in troubleshooting.

DNS Troubleshooting. Beyond server reachability and server/service status checks, troubleshooting of DNS resolution is a key function required to diagnose and resolve DNS issues. ISC provides a pair of configuration-checking utilities that are useful as a syntax check prior to loading network configuration or zone files.

CONFIGURATION FILE CHECK. The named-checkconf (144) command performs syntax checking of the named.conf file. The syntax of this command is

named-checkconf [-v] [-j] [-t *directory*] [*filepath*] [-z]

The command parameters are defined as follows:

- `-v`: prints the version of named-checkconf
- `-j`: read the journal file if it exists when loading a zone file
- `-t` *`directory`*: change root (chroot) to *`directory`* in order to process include directives
- *`filepath`*: path to the named.conf file, defaults to/etc/named.conf
- `-z`: load the master zone files as defined in named.conf to verify proper loading

ZONE FILE CHECK. The named-checkzone (144) utility provides syntax checking for a particular zone file. The syntax of this command is

named-checkzone [-v] [-j] [-d] [-q] [-c *class*] [-k *mode*] [–n *mode*] [-o *filename*] [-t *directory*] [-w *directory*] [-D] [*zonename*] [*filepath*]

The command parameters are defined as follows:

- -v: prints the version of named-checkzone
- -j: read the journal file if it exists when loading the zone file
- -d: enable debugging
- -q: quiet mode, 1=errors, 0=no errors
- -c *class*: specify zone class; default is IN
- -k *mode*: perform check-name checks on hostnames with *modes* of fail, warn (default) or ignore
- -n *mode*: check if NS records improperly use IP addresses as Rdata with *modes* of fail, warn (default) or ignore
- -o *filename*: write zone output to *filename*
- -t *directory*: change root (chroot) to *directory* in order to process include directives
- -w *directory*: change the current working directory to *directory* in order to process include directives
- zonename: domain name of the zone being checked
- *filepath*: path to the zone file

NAME SERVER LOOKUP. Among the most popular DNS diagnostic tools are nslookup (name server lookup) and dig (domain information groper). Nslookup is included with Windows DNS installations and both nslookup and dig ship with the BIND software distribution. Nslookup (145) is a simple utility that enables querying of a DNS server. Today, most administrators prefer dig, which provides much more detail and control over the query formulation, resolver configuration override, output formatting and more. To perform a single lookup using nslookup, simply type

nslookup *lookup-value [name server]*

where lookup-value is the host domain name or IP address to lookup and the *name server* is the server name or IP address to query. Additional options, specified below, may be included preceding the *lookup-value* with each option name prefixed with a hyphen (e.g., –timeout=5). Interactive mode for nslookup may be invoked by either entering nslookup with no arguments or entering nslookup followed by a hyphen, space character, and name server hostname or IP address like

nslookup- 172.18.71.105

Interactive mode enables entry of commands to formulate and perform queries. The following interactive mode commands are available:

- *host* [*nameserver*]: lookup the *host* on the specified *nameserver* or default server. This is similar to the command line format described above.

- `server` *`domain`*: lookup *`domain`* using the current [default] server and change the default server to that authoritative for the *`domain`* or the IP address specified in the *`domain`* field.
- `lserver` *`domain`*: lookup *`domain`* using the current [default] server and change the current server to that authoritative for the *`domain`* or the IP address specified in the *`domain`* field.
- `exit`: exits interactive mode.
- `set` *`option`*[=*`value`*]: set options to influence lookup behavior; the following may also be used in noninteractive mode by prefixing the option name with a hyphen on the nslookup command line.
  - `all`: displays current option values and current [default] server and host.
  - `class=`*`value`*: sets the Qclass within the query; valid *`value`*s are IN, CH, HS or ANY.
  - `[no]debug`: `debug` enables display of full response packet and `nodebug` disables this display.
  - [`no`]`d2`: `d2` turns on debugging and `nod2` turns off debugging display.
  - `domain=`*`name`*: sets the domain search list to domain *`name`*.
  - `[no]search`: `search` configures use of the domain search resolver configuration to append such domains to nonfully qualified searches containing at least one dot. `nosearch` disables use of search list.
  - `port=`*`value`*: sets the TCP/UDP port number to *`value`*; the default is 53.
  - `querytype=`*`value`*: sets the Qtype to a resource record type to query.
  - `type=`*`value`*: same as `querytype`.
  - `[no]recurse`: recurse issues a recursive query and `norecurse` does not.
  - `retry=`*`number`*: sets the *`number`* of query retries.
  - `timeout=`*`seconds`*: sets the number *`seconds`* to wait for a reply.
  - `[no]vc`: `vc` instructs nslookup to use TCP and `novc` to use UDP.
  - `[no]fail`: `nofail` sets nslookup to try the next name server if a SERVFAIL or a referral is received; `fail` does not try the next server.

Domain Information Groper. Dig (146) enables the formulation of a DNS query using standard DNS messages, emulating a resolver or recursive server. Dig provides granular control of the format of a query that can be sent to a DNS server in order to analyze the resulting response.

A common example usage of the dig command simply requests a resolution for a hostname:

```
dig @ns1.ipamworldwide.com A ftp-sf.ipamworldwide.com
```

This example would result in the issuance of an A record query for ftp-sf.ipamworldwide.com to the DNS server ns1.ipamworldwide.com. Valid possible arguments for the dig utility include

- `@server`: where *server* is the domain name or IP address of the DNS server to which to issue the query. If this parameter is not specified, dig will query the DNS servers listed in the client's resolver configuration.
- `-b address`: sets the source IP address of the query to *address*. This is useful for testing ACLs or views.
- `-c class`: class of the DNS resource record to query; the default is Internet.
- `-f filename`: enables issuance of successive queries as listed one per line in the specified *filename*. Format each line of the file as you would a dig command for the given query (without specifying "dig" on each line). This facilitates automated testing of a set of queries for critical resolutions in one step. Don't forget the boss' favorite resolutions.
- `-k filename`: signs the query and validates response signatures using TSIG, transaction signatures specified in *filename*. The TSIG key must match that defined in the DNS server's named.conf configuration.
- `-p port`: specifies the destination UDP (or TCP) port to use for the query; if not specified, the default DNS port, 53 is used.
- `-q name`: explicitly identifies the owner *name* to use in the query instead of using the "bare" `name` argument. That is, `dig –q sf-ftp1 = dig sf-ftp1`.
- `-t type`: explicitly identifies the query type to use instead of using the "bare" `type` argument. The default type is "A" unless `–x` is specified, indicating a PTR lookup.
- `-x address`: specifies a PTR lookup for the specified *address*. This option enables entry of an IPv4 or IPv6 address whereas if using `–t` with type PTR the name must be formatted as an .arpa. name.
- `-y [hmac:]name:key`: specifies an explicit TSIG key (instead of referring to a file with the `–k` option). The *hmac* field indicates key algorithm, HMAC-MD5 by default, the *name* field is the TSIG key name and *key* is the key itself. Care should be taken when using the `–y` option as the key can be visible from the output or command shell history. The TSIG key must match that defined in the DNS server's named.conf configuration.
- `-4`: send query using IPv4 transport.
- `-6`: send query using IPv6 transport.
- `name`: the owner name to query. Dig supports Internationalized Domain Names (IDN), so non-ASCII names may be specified.
- `type`: the query type to request in the query. The default type is A.
- `class`: the class of resource record to query. The default class is Internet.

- -h: prints help summary of the command; if no parameters are specified with the dig command, the help summary is provided.
- Query options: dig provides a number of options that can be specified to include or explicitly exclude query features. The plus sign is used to indicate each option and the no keyword indicates negation of the specified feature. The description of each option is written as if entered in the affirmative (without no). Note that a space is shown in this list between the optional no keyword and the option name where appropriate for readability. However, when entering respective commands, omit the space, for example, +notcp to negate the +tcp option.

Transport Options

- +bufsize=*bytes*: sets the UDP message buffer size to *bytes* bytes; valid values range from 0 to 65,535.
- +[no] fail: instructs dig to not try the next candidate server upon receiving a SERVFAIL result. This is the default behavior of dig, which is opposite that of a resolver.
- +[no] ignore: ignore any truncation resulting from a UDP query; normally such a UDP truncation scenario would lead to reissuing the query using TCP (which occurs when using +noignore) but using the +ignore setting instructs dig to not reissue the query using TCP.
- +[no] tcp: query using TCP; by default TCP is used for AXFR or IXFR queries and UDP is used for all other queries.
- +time=*time*: sets the query timeout to *time* seconds (default = 5, minimum = 1).
- +tries=*n*: sets the number of times, *n*, a UDP query will be sent in the absence of an answer (default = 3, minimum = 1).
- +retry=*n*: sets the number of times, *n*, a UDP query will be resent after the first query in the absence of an answer (default = 2, minimum = 1). To clarify, +tries specifies the total attempts while +retry specifies the number of attempts after the initial attempt.
- +[no] vc: query using TCP (vc = virtual circuit); by default TCP is used for AXFR or IXFR queries and UDP is used for all other queries.

Resolver Configuration Override Options

- +domain=*domainname*: sets the domain search list exclusively to the specified *domainname*.
- +ndots=*m*: specifies the number of dots (*m*) in the name to be considered fully qualified; that is, when fewer dots are entered in the name field, dig will append domain names specified in the domain or search parameter within the resolver configuration.

- +[no] search: enables domain search list processing based on the searchlist or domain directive specified in the resolver client.
- +[no] defname: Deprecated—equivalent to +[no] search.
- +[no] showsearch: display intermediate search results.

DNS Header Setting Options

- +[no] aaonly: sets the Authoritative Answer (AA) bit in the header of the query to indicate desire for an authoritative (noncached) answer.
- +[no] aaflag: equivalent to +[no] aaonly.
- +[no] adflag: sets the Authentic Data (AD) bit in the header of the query. This option is provided "for completeness" though it has no meaning. The AD bit is normally set by a server to indicate that the query response data has been verified via DNSSEC validation.
- +[no] cdflag: sets the Checking Disabled (CD) bit in the header of the query to instruct the queried DNS server to disable DNSSEC signature validation for this query.
- +[no] dnssec: sets the DNSSEC OK (DO) bit in the EDNS0 OPT record to indicate DNSSEC processing is desired.
- +edns=*version*: sets the EDNS version to *version*.
- +noedns: clears the EDNS *version* set with +edns.
- +[no] recurse: sets the Recursion Desired (RD) bit in the header of the query. Dig queries set the RD bit by default to request recursion except when the +nssearch or +trace options are specified.

Output Options

- +[no] all: displays the results in the default format; setting +noall suppresses all results.
- +[no] cmd: displays the first line of dig output showing the version of dig and the applied query options. This line displays by default.
- +[no] comments: dig normally displays results organized in DNS message format, organized by Header, Question, Answer, Authority, and Additional sections. These "sections" of the response along with blank lines are considered comments in the output that improve readability. Setting +nocomments suppresses these lines in the output. The output will still contain the query time, server, time stamp, and message size, though this can be surpressed using +nostats.
- +[no] identify: when used with +short, also displays the IP address and port number of the DNS server that provided each answer.
- +[no] multiline: displays complex resource record (e.g., SOA) results in multiline format; the default is to display each on a single line.

- +[no]nssearch: display the SOA record of the name servers authoritative for the zone specified in the name field (or –n parameter).
- +[no] short: display a terse answer. For example, when issuing an A query for a given name, only the resolved IP address(es) would be displayed.
- +[no] stats: displays the query time, responding name server, time stamp, and message size.
- +[no]trace: display the delegation path from the root servers to the authoritative name servers for the queried name. Dig will issue iterative queries to each server down the delegation path, displaying answers from each along the way. This is very helpful in identifying lame (broken) delegations in the domain tree.

DNS Message Options

- +[no] additional: displays the Additional section of the response (the default is to display the Additional section contents).
- +[no] answer: displays the Answer section of the response (the default is to display the Answer section contents).
- +[no] authority: displays the Authority section of the response (the default is to display the Authority section contents).
- +[no] besteffort: displays the contents of malformed messages (the default is to not display malformed responses).
- +[no] cl: displays the resource record class in dig results for this query.
- +[no] nsid: include the EDNS NSID request option to request the name server identity from the server.
- +[no] qr: displays the query as it was sent to the DNS server, organized by Header and Question fields by default.
- +[no] question: displays the Question section of the response (the default is to display the Question section contents).
- +[no] ttlid: displays the resource record TTL in dig results for this query.

DNSSEC Signature Validation

- +[no] sigchase: chase DNSSEC signature chains; requires dig be compiled with the -DDIG_SIGCHASE switch.
- +trusted-key=*filename*: identifies a *filename* containing trusted keys to be used with +sigchase; requires dig be compiled with the -DDIG_SIGCHASE switch.
- +[no] topdown: perform top-down validation when chasing DNSSEC signature chains used with +sigchase; requires dig be compiled with the -DDIG_SIGCHASE switch.

The dig utility for BIND 9 allows entry of multiple queries on a single command line simply by concatenating successive query name-type-parameters-options strings. For example, the following illustrates running a query for a PTR lookup for IP address 10.0.3.43 including display of the query along with a CNAME query for "ftp" while setting the resolver domain suffix to ipamworldwide.com.

dig –x 10.0.3.43 +qr ftp CNAME +domain=ipamworldwide.com

DHCP Troubleshooting. Testing DHCP transactions can be performed using DHCP client capabilities like ipconfig for Windows or ifconfig commands for Unix or Linux. These commands provide that ability to perform DHCP releases, renews, and set user class. For example, using `ipconfig` on a Microsoft Windows command line enables display of the IP configuration using the following arguments:

- `/all`: displays IP configuration information for each interface including
  - IPv4 address and subnet mask
  - Additional IP addresses including IPv6 addresses
  - MAC address(es)
  - Interface description
  - DNS domain suffix
  - Default gateway
  - DHCP server from which the lease was obtained along with dates/times the lease was obtained and that the lease expires
  - DNS servers for resolver configuration
  - WINS servers to query for NetBIOS lookups if configured
- Omitting the /all argument displays the IP addresses, subnet mask, domain suffix and default gateway only.
- `/?`: displays help in the form of a command summary.
- `/displaydns`: displays contents of the resolver's cache.
- `/showclassid` *`adapter`*: displays the user class configured for the specified interface adapter.

`ipconfig` also provides the following commands:

- `/release [`*`adapter`*`]`: Issues a DHCPRELEASE to release the lease for all or the specified interface adapter.
- `/renew [`*`adapter`*`]`: Issues a DHCPRENEW to renew all leases or that for the specified interface adapter.
- `/registerdns`: Issues a DHCPRENEW to renew all leases and updates DNS A record(s) directly (client to DNS server, not DHCP server to DNS).

- `/flushdns`: Clears the resolver cache.
- `/setclassid` *`adapter class`*: Sets the user class name for the specified interface adapter.

14.5 ACCOUNTING MANAGEMENT

Accounting management basically intends to keep everyone honest. Are those assigned addresses still in use? Are any unassigned addresses actually being used? Did the new subnet get provisioned on the router yet? Thus accounting management enables verification of successful configuration, as well as overall adherence to the IP addressing plan. Techniques for accounting management functions include discovery analysis of IP addresses, router subnets, switch port mappings, DNS resource records, and DHCP lease files.

Analysis of discovered information is necessary in order to compare this information with the IP inventory "plan of record." Such discrepancy reporting and comparison is difficult work, but provides a level of assurance of inventory accuracy. Without such a function, rogue users could access free service or otherwise infiltrate the network. In addition, planned network changes yet unimplemented may cause downstream process delays and violation of internal or external service level agreements on provisioning intervals.

14.5.1 Inventory Assurance

Each of the common IP management tasks we've covered so far relies on accurate IP address inventory to enable the allocation, deletion, and movement of blocks, subnets, IP addresses, and DHCP and DNS server configurations. Accuracy is absolutely essential for these address management tasks. But accurate inventory is also essential for general troubleshooting. Should a remote site be unreachable due to a network outage, it may be necessary to identify IP addresses, resource records, or other IPAM-related data for devices at the site. Only by maintaining an accurate IP inventory can such information be accessed when it may be needed most and when it cannot be obtained directly from the network.

In this section, we'll review steps you can take to assure the accuracy of your IP inventory. This includes controlling who can make certain changes to certain IPAM information, to discovering actual network data, reconciling the actuals with the inventory, and finally reclaiming address space.

Change Control and Administrator Accountability. As we've seen in reviewing these IP management tasks, a change in IP inventory often affects other network elements, including routers and DHCP and DNS servers. If different individuals or teams manage these different elements, it's a good idea to convene a planning or change control meeting periodically or as needed to review and schedule upcoming planned addressing changes. A little rigor can add some discipline to the process and keep those potentially affected by changes in the loop.

One way to help assure accuracy of IP inventory itself is to limit write access to the inventory to those whom are authoritative for and keenly knowledgeable of the IP addressing plan. Using a single password-protected spreadsheet that the one and only IP planner can modify is one approach to protecting the IP inventory from inadvertent or erroneous changes. However, for even modestly sized organizations, this approach is unwieldy. With the organization reliant on a single individual for the entire IP address plan, the individual must work around the clock and should he or she leave the organization, recovery of access to the inventory may be very difficult unless a successor is groomed in advance.

Support of multiple simultaneous administrators is a key feature of most IPAM systems on the market, and most allow some level of scope control so that certain administrators can only perform certain functions on certain devices or portions of the network. Make sure your chosen system supports administrator logging should you need to investigate "who did what" on the system.

As important as disciplined multi-administrator scoped access to the IP inventory is to delegating accountability, arbitrary changes to IP address assignments, DNS resource records, subnet addresses can be made outside of the scope of the IP inventory. For example, manual configurations can be mistyped, subnets can be provisioned on the wrong router interface, and client or DHCP updates to DNS can all contribute to IP inventory drift from reality. The IP inventory is a model of the IP address plan, and IPAM tasks rely on the accuracy of the plan. Therefore, it's prudent to take "pulse readings" from the IP network itself. Periodically polling and comparing the actual assignments on the network with the inventory is key to assuring inventory accuracy.

Network Discovery. A variety of methods are available to gather network actuals data, from ping, to DNS lookups, to SNMP polls. Pinging enables detection of an occupant of an IP address and provides a basic method to determine which IP addresses are in use for comparison with the respective portion of the IP inventory. Ping is very useful but be aware that some routers or firewalls will drop ping packets or even some devices can be configured to ignore pings. Setting up remote ping agents to perform local pinging on command can help avert the router/firewall traversal issue.

Nmap, freely available at insecure.org/nmap, is a particularly useful tool at the right price. It combines several discovery mechanisms to gather a variety of information from devices connected to the IP network, including ping sweeps, DNS lookups, and port scanning. When sweeping a subnet, nmap can perform these tasks in one command, issuing a ping to each address, looking up a corresponding PTR record in DNS, and attempting connections to various TCP and UDP ports to identify the device's operating system. From an IPAM perspective, ping results help identify IP address occupancy, DNS lookups help corroborate hostname-to-IP address mapping between DNS servers and the IP inventory and port scanning can provide additional information about the type of device occupying each IP address.

SNMP is another means of discovering IP inventory-related information. While most end devices like laptops or VoIP phones don't natively enable SNMP, most infrastructure elements like routers, switches, and servers do. Of particular interest

within router MIBs are the Interfaces, IpAddresses, and Arp tables. If your infrastructure devices support MIB-II, the interpretation of these tables *should* be consistent across different products. Just be aware of minor variations, even among different products from the same vendor. The information in these tables enables collection of the interfaces and subnets per interface provisioned as reported by the router. This provides useful validation of inventory in general, but can also be polled when in the process of allocating, moving, or deleting blocks and subnets.

Polling router ARP cache tables can provide a definitive mapping of MAC addresses to IP addresses on recent subnet communications. Even if a device refuses to respond to a ping, it must use the address resolution protocol (ARP) to formulate a layer 2 (e.g., Ethernet) frame in which to envelop its intended IP packet. As implied by the fact that this is cached information, it is transient and must be polled frequently.

Pinging of an IPv6 subnet is impractical given the sheer size of 2^{64} possible IP addresses on a /64 subnet. Polling of a router's Neighbor Discovery table, for example, ipv6NetToMedia SNMP MIB is a more effective means to perform IPv6 host discovery.

IP Inventory Reconciliation. Network discovery information provides a reality check on actual subnet allocations, IP address assignments and associated resource records. By comparing discovered information with the IP inventory database, discrepancies can be identified and investigated. While this comparison may require "eyeballing" the differences between the inventory spreadsheet and the discovery output, the effort can prove beneficial for several reasons. For example, database discrepancies can be identified that may be the result of

- *Incorrect Router Provisioning*. Incorrect subnet, mask, router interface, and so on.
- *Incomplete Router Provisioning*. Planned change not yet implemented.
- *Device Reachability Issue*. If a device should be at a given IP address and no response is received. This could result from a device outage, a transient outage (reboot), address reassignment, or network unreachability.
- *Incorrect IP Address Assignment*. Manually configured address is incorrect or device obtains a DHCP address from an unintended pool or address.
- *Actual IP Address Assignment*. For autoconfigured devices, the IP address is selected by the device. Also in some decentralized scenarios, the installer of a device on the subnet may select an IP address. Discovery can be used to update the IP inventory for these cases.
- *Incomplete IP Address Assignment*. All aspects of the assignment process, whether manual or DHCP, are incomplete. This issue is particularly applicable to manually assigned addresses where manual effort is needed to configure the assigned IP address, to detect autoconfigured devices and to update DNS.
- *Rogue Device Presence*. An unknown or unauthorized device has obtained an IP address. This provides an effective post-access control mechanism to complement and audit a network access control solution.

In addition to detecting discrepancies, analyzing discovery information can confirm completion of allocation or assignment tasks, as well as delete tasks. Discovery data is indispensible when moving blocks, subnets, and IP addresses. Since moves require allocation of the new address(es), movement, then deletion of the old address(es), confirmation of move completion is essential prior to deleting the old address(es) from the IP inventory. These addresses should not be deleted before the move completes so they are not unknowingly reassigned to other devices or subnets prior to their actual relinquishment.

In summary, network discovery is essential to assuring the accuracy of the IP inventory. It is also beneficial to monitoring provisioning or assignment progress and time frames, managing the completion of tasks requiring multiple related subtasks, and detecting incorrect assignments as well as potentially rogue devices.

14.5.2 Address Reclamation

Another benefit of network discovery and reconciliation discussed above is the detection of device reachability issues. If a server has been provisioned and has historically responded on a given IP address, but now no longer does so, such an event should stimulate further investigation. If there were no plans to move or decommission the device or there are no network problems reaching other devices on the subnet, the device may be suffering an outage, may be rebooting, may have been moved or disconnected, or may have been readdressed. If the server is providing critical services or applications, hopefully you're monitoring its status via a network management system* that can corroborate the outage theory and trigger corrective actions. If the IP address is discovered on the next attempt, perhaps it was simply rebooting. If it does not respond for the next *n* attempts, perhaps it is no longer physically (or at least electrically!) there. Unfortunately people don't always inform the IP planning team that a device has been removed or moved elsewhere, even in the tightest of organizations. A quick phone call to the site to check on the device's status may prove fruitful, but it's often difficult and time-consuming to identify the device's "owner" to verify status.

Nevertheless, the key point to assessing the possible fate of the device is that it may take multiple discovery attempts to determine if a device was there and no longer is, suffered a transient outage or disconnect, or was borrowed and has now been returned. Tracking a succession of discovery attempts may be difficult. A running log or spreadsheet can be used to log discrepancies or "missing" IP addresses as they are [not] detected. Reviewing this log over time may help determine if an IP address recorded as in use actually isn't.

In reviewing such a log, if a given IP address had been successfully discovered until a month ago, when it was last reachable after so many attempts, for example, 30, it may be confirmed as available for future assignment, or *reclaimable*. The concept of reclaim

* Or if the server is a DHCP or DNS server, it may be monitored via the IP management system.

entails identifying IP addresses that are denoted as in-use in the IP inventory, but are in reality not in-use, nor have they been in-use in recent history. Analyzing multiple discovery results provides a more robust sample set on which to base a reclaim decision, essentially deleting the device from the inventory and freeing it up for assignment to another device.

Besides providing robust confirmation of a device deletion from the IP inventory, reclaim may likewise be applied to subnets. When deleting a subnet, it's generally advisable to verify that all IP address occupants have been deleted and are no longer using IP addresses on the subnet*. Analyzing discovery results from all addresses on a given subnet can provide assurance that the subnet may be deleted. But like IP address reclaim, multiple sample sets provide more robust confirmation of the reclaimable disposition. Just keep in mind that you'll rarely see zero responses on a subnet, at least while it's still provisioned on a router interface, so you'll want to check successive discovery results ignoring routers, switches, and perhaps other device types.

14.6 PERFORMANCE MANAGEMENT

Performance management involves the monitoring functions of the IP management system and more importantly, the DHCP and DNS servers operating in the network. It's useful to track basic server statistics such as CPU utilization, memory, disk, and network interface input/output (I/O). Such monitoring enables tracking of the hardware's ability to support the DHCP and DNS (and any other services) running on the server. Trending analysis in this regard is beneficial as well to enable proactive planning of future hardware procurements to enable load distribution among more servers.

14.6.1 Services Monitoring

Monitoring of the DNS service helps assure adequate DNS horsepower to meet the demands for name resolution, and to help identify any exception conditions. BIND supports flexible logging of a variety of event types to a configurable set of output destinations or channels, including syslog, file, null, or stderr (the operating system's standard error output destination). Microsoft supports the DNS server event viewer with settable severity level reporting and counters for total queries/second received and responses/second sent. DHCP servers likewise provide logging to monitor overall service health and statistics, typically to a log file, syslog, or an event log.

These measures enable collection of performance data from the server's perspective. However, they don't convey the services performance as experienced by DHCP clients and DNS resolvers. Measuring client performance requires the remote issuance of a DNS

* Ignoring the router IP address's occupancy since it will typically identify itself on the subnet.

query or DHCPDISCOVER (or SOLICIT) packet and measuring the response time for receipt of a proper response*. This remote issuance could originate from services probes deployed in various locations to generate these "synthetic transactions," and measure and store response time results. Analyzing historical data from different probes can provide keen insight into DNS/DHCP services and network performance.

14.6.2 Address Capacity Management

Overall IP address capacity monitoring is another key performance management function within the scope of IPAM. Tracking address utilization from devices manually addressed, as well as those obtaining addresses via DHCP, enables proactive management of address space. Address allocations are initially based on estimated forecasts, which hopefully are accurate. Even when the forecast is perfect however, IP network dynamics due to employee movement, large events, subscriber growth, and unplanned address demands can consume the entire capacity of a subnet and its address pools. Periodic monitoring of utilization levels on pools and subnets, with historical tracking and trending can provide forewarning of a capacity crunch to trigger a supplemental allocation to expand capacity before it runs out.

Many DHCP server products enable monitoring of lease levels by command line, scripts or SNMP. Microsoft DHCP also provides a general 90% alerting threshold, providing notification should an address pool reach 90% capacity. Other servers and IP management systems provide similar or additional alert threshold definition and application. It's usually better to be notified by an IP management or network monitoring system than by irritated customers or end users attempting to access the network.

14.6.3 Auditing and Reporting

Most management systems in general provide some level of auditing of "who did what" and varying levels of reporting. These functions, which could just as easily be categorized under Accounting Management, enable administrators to track and troubleshoot activity and to convey status information in report format. Auditing of IP address usage, that is, who had a given IP address at a certain point in time is valuable information when troubleshooting a network issue or investigating potential illicit activity. Likewise, if you are attempting to track the history of IP address occupancy for a given device, reporting by hardware address is also beneficial.

Performing such auditing without an IP management system may be difficult except for the smallest of networks. Iterative dumps of DHCP lease data to track dynamically addressed clients over time are necessary. The ability to search for a given IP address requires access to a single (or two if failover or split scopes is in effect) DHCP server's lease history, while the search by hardware address necessitates searching across all DHCP servers, assuming the device is capable of mobility.

* As mentioned in the Fault Management section, the absence of a response may indicate a services outage and should be investigated if it persists.

Common reports of interest for IP address planning include the following, though your system may provide different or additional reports.

- *Address Utilization Report*. By pool, subnet, block, rollups through hierarchy.
- *Address Assignment Report*. Summary of assigned addresses by subnet or block as current snapshot and/or history.
- *Address Discrepancy Report*. Highlights of discrepancies between the IP inventory and discovered IP address information.
- *DHCP Performance Report*. Summary and details of DHCP protocol messages by type and/or client and server key metrics summary.
- *DNS Performance Report*. Summary and details of queries by type, by querier, by question and server key metrics summary.
- *Audit Reports*. Administrator activity, by subnet, by IP address, by hardware address, by resource record, by server.

14.7 SECURITY MANAGEMENT

Several IP management vendors have attempted to join the "NAC" bandwagon, enabling IP planners to restrict IP address assignments only to valid devices or users, as determined by DHCP MAC address filtering or user login. Such a solution needs to provide exception reporting and ideally alerting, should a number of access attempts by a device or user fail. One failed access attempt can likely be attributed to mistyping, but several failures may indicate an attacker attempting to crack the system to access the network. Of course, if the attacker knows the subnet address, he/she could manually configure an IP address to access the network, bypassing DHCP. This process is discussed in detail along with alternative approaches in Chapter 8.

Chapters 8, 12 and 13 address securing the information on and transactions with DHCP and DNS severs, respectively. Securing the IP inventory and DHCP and DNS configuration data is also important, as this information is critical and should be protected from sabotage. Protecting IP data requires administrator access controls, enabling at least password-protected access to the information. If your organization has more than two or three administrators, a more sophisticated administrator security approach may be warranted with the use of an IP management system. Most systems go beyond minimal password entry to enable scoping of administrator access by system function, by portions or certain elements of the network, by access type, for example, super user versus read-only access and more. The sophistication of access controls will be driven by the number of administrators, their respective roles and responsibilities, and organizational security policies.

14.8 DISASTER RECOVERY/BUSINESS CONTINUITY

Business continuity practices seek to maintain the operation of the enterprise in the face of a major outage. A major outage or "disaster," implies that the sheer magnitude of the

outage goes beyond a handful of servers or network devices. Automated and manual procedures must be documented in advance to reconfigure or redeploy resources to maintain operation of the network and applications, or at least the critical services and applications. We've discussed a variety of approaches for deploying and configuring redundant DNS and DHCP services in Chapters 7 and 11. Once deployed and configured, redundancy features should provide network services continuity in the event of an individual server outage. On top of these technologies, deployment of DHCP and DNS appliances may provide an added layer of availability. While typically providing intrasite hardware redundancy, this deployment model is not normally considered for disaster recovery solutions due to co-location requirements. Appliance redundancy nevertheless provides a viable redundancy option especially for critical servers, such as master DNS servers.

Business continuity of IPAM operations will likely require deployment of multiple IPAM databases. Deployment of multiple active databases or primary/backup configurations will depend on your selected vendor. Vendors implement a wide variety of approaches to facilitate redundancy such as full database copies and transfer, multimaster databases that require some level of network partitioning, to deployment of database replication technologies using storage area networks or SQL or LDAP replication capabilities. Operations tasks required to perform a disaster recovery will likewise vary per vendor.

Evaluation of vendor disaster recovery capabilities will depend on your business objectives, budget and polices, but three key questions should be considered:

1. Is the IPAM database involved in name resolution or address assignment? For example, some systems route dynamic updates from DHCP to the IPAM database for uniqueness checks prior to routing to DNS. If the IPAM database is in such a "critical path," redundancy and high availability are paramount.
2. How frequently do your administrators make changes to IPAM data? The more frequent the rate of change, the more data changes may be lost between data synchronizations between the primary and backup database(s). A daily database backup may be acceptable in cases where changes are made infrequently whereas a subdaily or transactional replication process may be required for high rate-of-change environments.
3. What is the process to perform invocation of the backup system? Some vendors provide a relatively simple recovery procedure, while others require more manual intervention. Some small level of manual intervention is probably desirable given the broad impact of a disaster recovery failover. Intermittent issues may lead to false positives and potentially disruptive failovers, so initiating the failover manually can eliminate a disaster caused by the solution. Hopefully disasters occur infrequently if never, but when needed, the failover process should be executable by staff on hand within time constraints defined by policy.

These basic questions are interdependent. If the answers to questions 1 and 2 are "yes" and "frequently," respectively, then the answer to question 3 should probably be "single step" or at least "very streamlined."

14.9 ITIL PROCESS MAPPINGS

The IT Infrastructure Library* is a documented set of best practices for use by an IT organization desiring to manage, monitor, and continually improve IT services provided to the enterprise organization. ITIL was developed by the U.K. Office of Government and Commerce, and its IT service-oriented approach has been deployed by a number of organizations. The most common drivers for ITIL implementation include

- Costs reduction of IT services delivery to the organization.
- Risk management through disciplined planning and evaluation of potential service-affecting changes.
- IT service level consistency and improvements.
- Efficiencies in utilizing documented processes and continual improvement.

Many of the functions within these process areas are identical or similar to those discussed earlier in the chapter, so we'll simply summarize these within their respective mappings to ITIL process areas.

14.9.1 ITIL Process Areas

ITIL processes are split into two areas: service delivery and service support. Service delivery involves the planning, development, and deployment of IT services, while service support entails service operations in managing service levels and support. The service desk is a third process area that integrates with both service delivery and service support to provide a unified interface for IT to the rest of the organization. ITIL version 3 builds upon these sets of processes, with a couple additions and functional splits.

Service Delivery. *Service level management* is a service delivery process area that encompasses the specification of service levels for various services provided by the IT organization. This is akin to a service level agreement. Service level management also includes measurement of service delivery against these specifications to monitor adherence to and measuring the level of service that IT provides.

From an IPAM perspective, service level management can involve definition and measurement of the level of service provided to those requesting IPAM-related services, whether it be end users requesting an IP address or the business needing to open a new retail office. Treating the end user or the business in these cases as customers, this process seeks to gauge whether service delivery is meeting defined service levels, such as timeliness of completion of these requests. Automating IPAM-related service delivery, whether solely IPAM impacting or involving IPAM as part of a larger IT service such as

* Details are available at the ITIL web site, http://www.itil-officialsite.com/home/home.asp (168). Functional mappings in the section are based on those initially discussed in (174).

VoIP deployment, facilitates timely and accurate services delivery. An example metric is the time frame by which a subnet or an IP address will be assigned.

Financial management naturally includes accounting, similar to accounting management in the FCAPS model, though the financial management area addresses actual dollars and cents as well. This process area also deals with any chargebacks or cost allocations for certain departments under an IT funding or cost allocation model.

Some firms do implement cost chargebacks for IP address usage. In such a scenario, the financial management processes would need to account for tracking of IP address usage along with the corresponding user and chargeable entity (e.g., department). Depending on the billing or chargeback cycle, this IP address use information will need to be stored for the current cycle and more to enable archiving or dispute resolution. Audits and history data in your IPAM system can also be a big help in justifying cost allocations.

Capacity management simply involves assuring adequate IT resources of the proper type are available for the business to conduct its work. Considering the application of this concept to IPAM, certainly IP address capacity management springs to mind, but one should also consider DHCP and DNS server load capacity. In the former case, capacity management requires monitoring of addresses and address pools to provide enough IP addresses for employees to get an address and access the network. Monitoring for trends is helpful, and enabling alerting for low pools is also recommended for tightly allocated networks. Of course, given the magnitude of IPv6 address space, this will likely not be an issue for IPv6 space for the foreseeable future.

With respect to server capacity management, monitoring each server's network, memory and CPU utilization over time can provide insights into its performance. Such performance tasks may in fact be required as a linkage to service level management in terms of percentage of transaction completion (lease or resolution) as well as response times. Regardless, excessive loads on servers can have detrimental impacts on the availability of DNS and DHCP services, so server monitoring and perhaps even probe-like transactional monitoring can provide effective measures of service levels and capacity.

Availability management is a service delivery process area focused on making sure IT services are available to end users. High availability, a common goal for applications including DHCP and DNS, requires deployment of redundant configurations and the ability to leverage these configurations to provide continuous service in the face of a component outage.

As discussed in Chapters 7 and 11, deployment of redundant appliances can provide localized clustering, which with implementation of DHCP failover or split-scopes and multi-server DNS deployments, provides an additional layer of redundancy. Redundant IPAM database deployments through LDAP or replicated relational databases can also assure availability of the IPAM application for managing IP space. Monitoring of the availability of each of these redundant components enables proactive detection of outages to facilitate rapid outage resolution (mean time to repair) while redundant components shoulder the load.

Continuity management is related to availability management in that it deals with providing continuously available services. For example, in the event of a disaster, this

process area would require a disaster recovery plan be in place. As discussed in the Business Continuity section earlier, a variety of strategies are available based on the criticality and scope requirements of the organization for particular DHCP/DNS servers and IPAM systems.

Service Desk. Serving as the interface to the user community, the service desk filters input to the IT organization for incident reporting, change requests and even new service requests. It serves to filter and direct user requests or problems to any one of the other ITIL areas, providing end users with a helpdesk function.

The policies and culture of the organization will drive whether the service desk performs traditional "level 1" support only by logging troubles with subsequent follow-up, or higher support levels up front to perform a level of diagnosis. In the case of level 1 support, little more is needed than a ticketing system with the ability to assign tickets to those responsible for other process areas depending on the caller's issue. A service desk staffed to perform some trouble diagnosis will require access to status monitoring tools to try to "see what the caller sees" with respect to the issue.

For IP address or name resolution-related calls, providing service desk personnel access to IP inventory information may prove beneficial. For instance, if a person located in the Philadelphia Headquarters office is not able to get an IP address, the service desk needs to know the address plan for Headquarters in order to focus the problem and trouble resolution process on that particular subnet, associated routers or DHCP/DNS servers.

The service desk is the interface not only for trouble reports, but for change requests, such as IP subnet or address assignments. Providing service desk personnel with basic access to the IPAM system to request such changes, or better yet, enable end users themselves to register such service requests to an automated IT portal can increase end users' satisfaction with IT services through rapid fulfillment.

Service Support. *Incident management* is a service support process area that involves tracking and resolving incidents. In ITIL version 2, it also deals with change requests, whereas in version 3, these are split into separate process areas. As described above, the service desk receives incident and change requests from the end user community directly. Through network monitoring IT can also detect and troubleshoot network issues proactively. Regardless of the means of detection for a given incident, access to IP inventory data is indispensible to troubleshooting and incident resolution. In addition, monitoring of server states with thresholds, alerts, logging information, and audits can provide a head start to incident detection and management.

Service requests can be handled by service request tickets initiated via the service desk or IT web portal as described in the Service Desk section.

Problem management calls for the tracking of known problems and resolutions in a known problem database. If someone calls into the service desk with an incident, for example, it could get bumped over to problem management to identify whether this incident has been reported and addressed in the past. If so, the defined resolution path can be followed to quickly troubleshoot and resolve the issue.

While IPAM systems traditionally don't store problem histories with resolution annotations, some can provide a database of problem information through logging history, as well as inventory change audits. Network management system integration through APIs can provide a holistic view of problem history by providing IPAM data through the API to a trouble ticketing system for example. IPAM is a key part of the overall network or IT service management approach, but it's not comprehensive; no system is. Having that integration is a key to having a holistic view of the problem management scope.

Configuration management within ITIL is similar to the FCAPS configuration management functionality in terms of identifying, recording, and controlling configuration parameters affecting IT services. And as we discussed extensively in this chapter, configuration management functions are a large focus area of IPAM processes. This includes configuring new address pools from a DHCP perspective, zones and resource records in DNS, IP addresses for subnets on routers, and so on, which all fall into the realm of configuration management. The IPAM database can be considered a configuration management database (CMDB) component of an IT's confederation of CMDBs for network configuration inventory.

Administrator controls need to be considered for organizations with more than one IPAM administrator to ensure that changes to DHCP and DNS configurations are done with the appropriate scope and permissions. For instance, you may want administrators to be able to make changes, but not actually deploy them on the DHCP and DNS servers—restricting that function to a higher level of administrator. On the back end, audit information is key for accountability tracking and reporting.

Possessing accurate IP configuration information is needed to provide a solid foundation on which future configuration changes can be planned. A corollary requirement leads to the necessity of validating that inventory against network actual data. IP inventory tracked on a spreadsheet is fine but requires constant updating. Having the ability to collect information from the network and then compare it with the plan is very important. Audits go hand in hand with inventory information collection. Arming the service desk with this information can provide a solid first line of defense for addressing calls immediately, or to at least moving them through the process more quickly.

Change management provides controls on the implementation of changes in the IT infrastructure. This involves assuring that all affected parties are in agreement with respect to the scope and implementation timing of the proposed change. In terms of IPAM, the scope of change management commonly affects IPAM components, such as the addition of an address pool, deployment of a new DHCP/DNS server in the network, or upgrading a server to a new software version. Basically, anything affecting any part of the infrastructure, whether it's physical or software or even underlying appliance operating system, falls under the change management process, which seeks to assure all appropriate approvals are in place and corresponding backout plans are available.

Release management is a service support process area that provides controls on deployed releases for hardware and software versions, not only for operating systems, but also for applications and appliances. This process area is responsible for making those versions available and accessible on the IT network and making sure there's an authorized set of releases and versions available that can be deployed appropriately.

Release planning and upgrade management for DHCP and DNS servers via a central location can be a big timesaver. The alternative requiring on-site upgrades of operating system, patches, and application software is costly and time-consuming. Release management of the IPAM system also falls within this category.

CONCLUSION

FCAPS and ITIL are similar in that they both advocate documented processes with disciplined execution. The importance of IPAM to an organization should drive application of FCAPS or ITIL principles to the practice of IPAM within the organization. This chapter has presented a detailed perspective of key IPAM process steps in the context of an FCAPS framework, as well as a suggested mapping to ITIL.