3

IP ADDRESS ALLOCATION

In this chapter, we will begin describing the technology and applications that serve as the foundation of the practice of IP address management. In addition, we will illustrate the technology and applications by way of example. Thus beginning with the fundamentals of IP address allocation, we'll incrementally apply each new concept to a fictitious organization called International Processing and Materials (IPAM) Worldwide (play on words intended!). IPAM Worldwide′s basic organization consists of a global headquarters in Philadelphia and three major geographic headquarters spanning the world, in Europe at Dublin, in North America at Philadelphia, and in Asia at Tokyo. IPAM Worldwide has about 17,000 employees and 24 distribution centers, which also serve as branch offices, and an additional 37 offices functioning solely as branch offices. Figure 3.1 illustrates a basic location spreadsheet, highlighting each continental headquarters, and corresponding distribution centers and branch offices.

The deployment of the IP network will primarily be driven by where the users of the IP network are located per the sites listed in Figure 3.1, by the number of users at each location, by the variety of user requirements for access to information resources such as internal applications and the Internet, and by the variety of administration requirements for managing the IP network from security to auditing. Because of the variety of inputs

IP Address Management: Principles and Practice, by Timothy Rooney

IPAM Worldwide Global Locations				
Core Sites	**Region**	**Regional Site**	**Distribution Centers**	**Branch Offices**
Philadelphia	HQ—Corporate	Philadelphia		
Philadelphia	HQ—North America	Philadelphia		
	N. America—East	Norristown	Toronto Nashua Newark Baltimore Pittsburgh Charlotte Atlanta	Providence Quincy Albany Manhattan Ocean City Reston Richmond Charleston Montgomery
	N. America—Central	Kansas City	Chicago Des Moines Memphis New Orleans Mexico City	Lisle Indianapolis Topeka Houston
	N. America—West	San Francisco	Denver Vancouver Phoenix	Calgary Albuquerque Salt Lake City Boulder Edmonton Sacramento Anaheim
Dublin	HQ—Europe	Dublin		
	Europe—West	London	Amsterdam Paris	Manchester Madrid Lyon Lisbon
	Europe—South	Rome	Rome	Nice Milan Athens
	Europe—East	Berlin	Munich Moscow	Vienna Prague Budapest Kiev
Tokyo	HQ—Asia	Tokyo	Tokyo Beijing Singapore Auckland	Seoul Osaka Singapore Manila New Delhi Sydney

Figure 3.1. IPAM Worldwide global locations and offices.

related to individual business needs, the IP network of any one organization generally looks somewhat different from that of any other. However, the techniques we discuss should be broadly applicable across a wide variety of networks, including yours.

The IT team at IPAM Worldwide has decided to deploy a high-speed backbone or core network among the organizational and geographic headquarters. Emanating from each regional headquarters office is an intracontinental wide area network (WAN) interconnecting each of the region's retail, distribution, and branch offices. Building on this basic two-layer hierarchy of core and regional networks, each branch network is further divided by geographic region. For example, within North America, they've divided the administration into three subregions: east, central, and west, and then further by major distribution center and branch office site. Likewise, the Europe region has been subdivided into west, south, and east regions.

Following this topology, the IT team has decided to mimic this structure with respect to address space, as we'll see next. Hence, a core network interconnects the regional headquarters sites, and each regional headquarters serves as an intermediary between its corresponding regional network and the core network. Each regional network interconnects its respective distribution centers and branch offices within the region. From an organizational perspective, each region has its own IT team that would like to manage its own space and associated DHCP and DNS server configuration. Figure 3.2 depicts the high-level IPAM Worldwide network topology design.

In terms of IP address space allocation, IPAM Worldwide will deploy a 10.0.0.0/8 network from the RFC 1918 private address space. Public address space, 192.0.2.0/24, has been obtained from an ISP (we'll discuss where this space comes from and ISP public address space allocation and policies later in this chapter). This public space will be allocated for Internet facing devices like web servers, email gateways, and VPN gateways for partner connections and remote employees. In addition, a portion of the public address space is reserved for deployment as a public address pool on a network address translation (NAT) firewall facing the ISP. As we introduced in Chapter 1, a NAT can be configured to perform private-to-public address conversion automatically in order to enable privately addressed (internal) hosts to access the Internet.

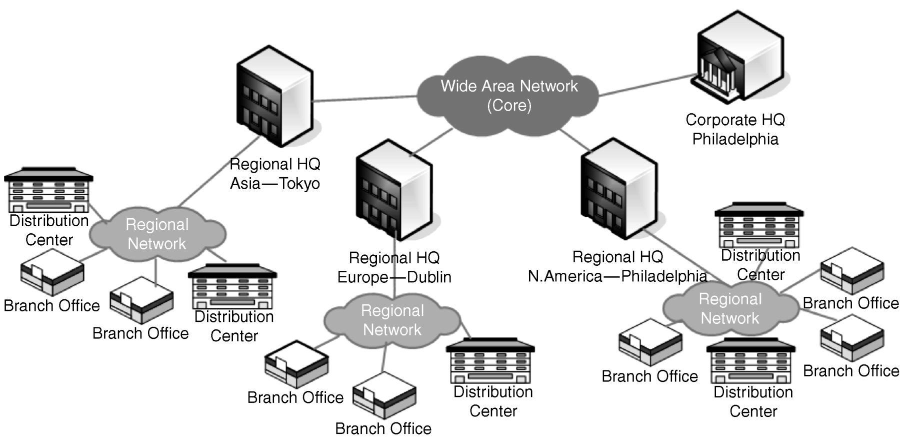

Figure 3.2. IPAM Worldwide network topology (partial).

3.1 ADDRESS ALLOCATION LOGIC*

Effective IP address allocation requires diligent planning and, ideally, accurate forecasting. Knowing the IP address space requirements for every level of the network hierarchy enables optimal allocation of address space to fully meet the address capacity needs while minimizing address space waste. Of course in reality, having an accurate long-term IP address forecast is a rare luxury. Business needs drive constant change with new sites opening, some closing or moving, new IT initiatives like rolling out voice over IP, and even mergers. Beyond these strategic events that can usually be planned for proactively, organizational dynamics can drive shorter term perturbations in address capacity requirements. For example, perhaps a regional organization is conducting a wildly successful customer event, driving a surge in IP address demand, or a new project is causing a shift in IP address needs due to temporary colocation of project resources.

The bottom line is to do your best in mapping out high-level address capacity needs, add some additional "insurance" address space to the extent possible, then proactively monitor address utilization as a feedback loop to ensure the addresses allocated are being effectively utilized given the short- and long-term address-affecting events. A key function in the management of IP space, proactive monitoring can trigger the allocation or movement of address space to where it may be needed more urgently.

Of course, the intensity of proactive monitoring will be directly proportional to the utilization of the address space. If many of your networks are above 90% utilization,† you may need to monitor their utilization hourly or at least multiple times per day. Networks with utilization below 70% may require monitoring checks a couple times a week. Ideally, you can define thresholds and alert conditions within a monitoring or IP address management system to alleviate the need to constantly monitor your networks manually, and have the associated management system collect the information and alert you to a particular capacity utilization condition.

Beyond capacity needs, another important consideration is the allocation of IP address blocks in a hierarchical manner such that address space "rolls up" to the highest level efficiently. This practice is critical to maximizing route aggregation to reduce routing protocol traffic and routing table overhead. It's important then to consider routing topology when allocating space.

A third consideration is a more recent phenomenon: allocation per application. Due to latency or quality of service (QoS) requirements for certain applications such as voice over IP, which generally requires low latency on the order of tens of milliseconds versus data, which tolerates multisecond time latency, some network planners implement routing treatment based on application. One way to perform this is to carve out a portion of the overall address space for voice treatment with higher priority queuing for example, while separating this from the data space. Other "special need" IP applications may require further address delineation.

* Allocation logic and examples are based on analogous content in Chapter 6 of Ref. 11.

† Blanket percentages aren't always the best trigger for action, especially when you have different sized subnets throughout the organization. Ninety percent utilization of a subnet with 10 addresses would certainly be of higher urgency than would one with 1000 addresses.

3.1.1 Top-Level Allocation Logic

To illustrate address allocation concepts, let's apply them to IPAM Worldwide's private address block, 10.0.0.0/8. When performing top-level allocations such as this, keep in mind not only the capacity required in terms of IP addresses but also the number of subdivisions or hierarchy layers that may be ultimately necessary. In the case of IPAM Worldwide, we will define our address hierarchy layers as follows:

- Application
- Continental or core layer
- Regions
- Sites or buildings

Thus, our top-level allocation will divide our address space by application. Each application-specific allocation will then be allocated at the core router or continental level, then by region, and finally by office. Since we have four layers of allocation hierarchy, we will have to allocate along nonoctet boundaries. So let's look at this from both a CIDR network notation and the corresponding binary notation.

The binary representation of this network is shown below. The network portion of the address, whose length is identified by the /8 notation is highlighted as bold italics, while the local portion is in plain text.

Private Network	10.0.0.0/8	***00001010*** 00000000 00000000 00000000

Before we allocate this space across the organization, let's assume that IPAM Worldwide is planning to roll out voice over IP in the near future and additional IP services later on. Let's take the next four bits of our network address and allocate equal-sized /12 networks. This would provide $2^{(12-8)} = 16$ potential high-level allocations, while providing $2^{(32-12)} > 1$ million IP addresses per allocation. Thus, we'll allocate a /12 each for the infrastructure address space, a /12 for the voice over IP address "subspace," and a /12 for the data subspace. This allocation is illustrated below with the bold italic bits once again representing the network (network + subnet) portion and normally formatted bits representing the host bits.

Private Network	10.0.0.0/8	***00001010*** 00000000 00000000 00000000
Infrastructure	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
Voice	10.16.0.0/12	***00001010 0001***0000 00000000 00000000
Data	10.32.0.0/12	***00001010 0010***0000 00000000 00000000

3.1.2 Second-Level Allocation Logic

This initial application-level allocation reformats our original monolithic /8 address space into three /12 spaces aligned per application. The decision to use /12 at this level is a trade-off between the number of first-level allocations and the number of addresses available per allocation. If we had decided to allocate /11s, we would have ended up with 8 /11s total, each with over 2 million IP addresses. In IPAM Worldwide's case, having more top-level blocks available for future allocation was more of a concern than managing capacity per block, given 1 million IP addresses per /12 application block. If a particular allocation becomes exhausted, we can allocate an additional /12 block.

Block sizing decisions for second and subsequent level allocations should generally employ different logic. Instead of trading off allocation size with the number of equal-sized allocations, an optimal allocation strategy should be used. This optimal strategy entails successively halving the address space down to the size required. The key reason for this approach is that it enables you to retain larger blocks of unallocated address space as available for larger requests and alternative allocations.

If you have ever endured a company merger, you may have encountered a situation like the following that illustrates the optimal allocation motivation. Let's say IPAM Worldwide acquires a company and the network integration strategy requires an allocation of 250,000 IP addresses to the new division. To minimize confusion (and to exude networking mastery over the rival IT organization), IPAM Worldwide desires to allocate a single /14 to support 262,142 addresses.

If we've optimally allocated our address space, we may happen to have a /14 readily available (IP address master!). If we had taken a uniform approach of allocating /16s everywhere, we may be lucky to identify four contiguous /16s that comprise a /14 (lucky amateur!). If we cannot identify four contiguous /16s, we may have to assign four noncontiguous /16s; this adds four times the overhead to routing tables and routing protocol update entries (four /16s versus one /14 – rookie!). With successive halving instead of the uniform single-sized approach, a /14 is more likely to be readily available for assignment. Let's look at how this works.

If we start with our 10.0.0.0/12 infrastructure block and halve it, we end up with two /13 blocks as illustrated below. Thanks to binary arithmetic, note that associating the next "host" bit with the network enables halving of the original network. Note that the 10.0.0.0/12 network no longer exists, so we've grayed it out to illustrate this; it has been split into our two /13 networks.

Original Network	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
First half	10.0.0.0/13	***00001010 00000***000 00000000 00000000
Second half	10.8.0.0/13	***00001010 00001***000 00000000 00000000

Next, let's halve the "first half" above, leaving the 10.8.0.0/13 block available for future allocation or subnetworking for infrastructure applications (or acquisitions!). We extend the network portion of the address now to the 14th bit to halve the 10.0.0.0/13 to yield two /14s as below. Note that as with the 10.0.0.0/12 network, the 10.0.0/13 network no longer exists as an entity and also has been grayed out. It has been split into the two /14s as shown below. However, the 10.8.0.0/13 network is available to the organization for further allocation as needed.

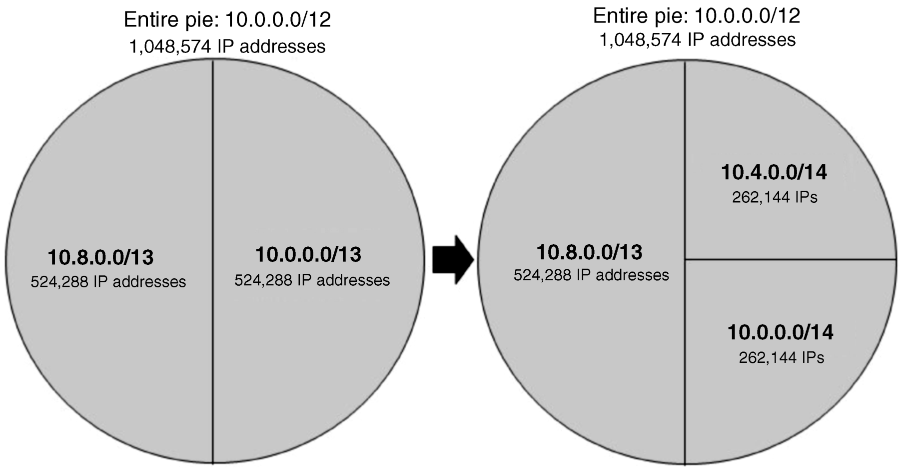

Figure 3.3. Pie chart view of address allocations (Based on [11] and [166]).

Original Network	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
Original first half	10.0.0.0/13	***00001010 00000***000 00000000 00000000
First /14	10.0.0.0/14	***00001010 000000***00 00000000 00000000
Second /14	10.4.0.0/14	***00001010 000001***00 00000000 00000000
Second half	10.8.0.0/13	***00001010 00001***000 00000000 00000000

One way to visualize this halving process from an overall allocation perspective is to view the address space as a pie chart as shown in Figure 3.3. If our entire pie represents the base network, 10.0.0.0/12, then we cut it in half to render two /13s as shown on left of Figure 3.3. We can then leave one of the /13s as “available” (left half) and slice the other /13 (right half) into two /14s as shown in the right half of Figure 3.3.

Continuing to apply this logic down to a /16, we end up with the following:

Original Network	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
First half (/13)	10.0.0.0/13	***00001010 00000***000 00000000 00000000
First /14	10.0.0.0/14	***00001010 000000***00 00000000 00000000
First /15	10.0.0.0/15	***00001010 0000000***0 00000000 00000000
First /16	10.0.0.0/16	***00001010 00000000*** 00000000 00000000
Second /16	10.1.0.0/16	***00001010 00000001*** 00000000 00000000
Second /15	10.2.0.0/15	***00001010 0000001***0 00000000 00000000
Second /14	10.4.0.0/14	***00001010 000001***00 00000000 00000000
Second half (/13)	10.8.0.0/13	***00001010 00001***000 00000000 00000000

As each “first” block is split, it creates two networks of network mask length of 1 bit longer than the original block. Now that we’ve performed this split, we have two /16 networks: 10.0.0.0/16 and 10.1.0.0/16 as derived above. We are also left with one /15, one /14, and one /13 shown below our two highlighted /16 blocks. These exist because the "first" set of networks were successively sliced into half, yielding a "first" network that was further subdivided and a "second" network that could be preserved for additional future allocations or assignments. The smallest "first" network, 10.0.0.0/16, is the one we can allocate as it is of the required size.

10.0.0.0/16	***00001010 00000000*** 00000000 00000000
10.1.0.0/16	***00001010 00000001*** 00000000 00000000
10.2.0.0/15	***00001010 0000001***0 00000000 00000000
10.4.0.0/14	***00001010 000001***00 00000000 00000000
10.8.0.0/13	***00001010 00001***000 00000000 00000000

But IPAM Worldwide requires a third /16. From which block shall we allocate this? In keeping with our recommendation to retain larger blocks, we'll take our next available network of the smallest size. In our case, from the listing above, the 10.2.0.0/15 network is available for further allocation. If we split this /15 into two /16s we have 10.2.0.0/16 and 10.3.0.0/16. We can then assign the former of these two networks as we illustrated earlier and retain the latter network as available for future assignment. The resulting pie chart is illustrated in Figure 3.4, with the allocated space.

Note that we still have many large blocks available for further allocation or assignment. Only the darker shaded wedge of the pie comprising our three /16 networks has been assigned. In relating the successive splits in the table above to the pie chart, while each "first" half block was either assigned or divided into further allocations, it yielded a corresponding "second" half block that is still free or available. Thus, the

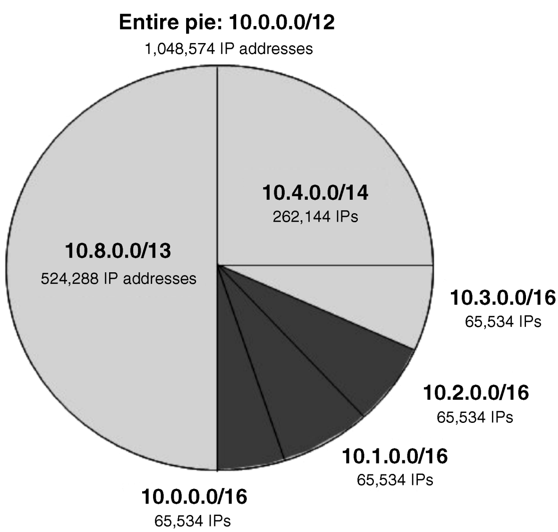

Figure 3.4. Allocation of three /16s from /12 space [Based on (11) and (166)].

resulting address allocations for IPAM Worldwide based on this initial allocation are as follows:

Original infrastructure (IS) block	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
Free IS block	10.8.0.0/13	***00001010 00001***000 00000000 00000000
Free IS block	10.4.0.0/14	***00001010 000001***00 00000000 00000000
N. America IS block	**10.0.0.0/16**	***00001010 00000000*** 00000000 00000000
Europe IS block	**10.1.0.0/16**	***00001010 00000001*** 00000000 00000000
Asia IS block	**10.2.0.0/16**	***00001010 00000010*** 00000000 00000000
Free IS block	10.3.0.0/16	***00001010 00000011*** 00000000 00000000

Following similar logic with the data and voice top-level address allocations, 10.16.0.0/12 and 10.32.0.0/12 respectively, we can derive the following allocations:

Original Voice block	10.16.0.0/12	***00001010 0001***0000 00000000 00000000
Free voice block	10.24.0.0/13	***00001010 00011***000 00000000 00000000
Free voice block	10.20.0.0/14	***00001010 000101***00 00000000 00000000
N. America voice block	**10.16.0.0/16**	***00001010 00010000*** 00000000 00000000
Europe voice block	**10.17.0.0/16**	***00001010 00010001*** 00000000 00000000
Asia voice block	**10.18.0.0/16**	***00001010 00010010*** 00000000 00000000
Free voice block	10.19.0.0/16	***00001010 00010011*** 00000000 00000000
Original Data block	10.32.0.0/12	***00001010 0010***0000 00000000 00000000
Free data block	10.40.0.0/13	***00001010 00101***000 00000000 00000000
Free data lock	10.36.0.0/14	***00001010 001001***00 00000000 00000000
N. America data block	**10.32.0.0/16**	***00001010 00100000*** 00000000 00000000
Europe data block	**10.33.0.0/16**	***00001010 00100001*** 00000000 00000000
Asia data block	**10.34.0.0/16**	***00001010 00100010*** 00000000 00000000
Free data block	10.35.0.0/16	***00001010 00100011*** 00000000 00000000

The only remaining step at this core level is to allocate infrastructure space for the core routers themselves. The core network is after all a network requiring an IP subnet address and it lies "above" our intercontinental allocations. For this subnet, we'll carve out a /26 subnet. This sized subnet provides 62 host addresses, which provides sufficient capacity for growth. Let's allocate this from the smallest free infrastructure block, 10.3.0.0/16. Following similar logic just applied, we allocate the 10.3.0.0/26 network to our core backbone network. We have several free blocks available for future allocation:

Original IS block	10.3.0.0/16	***00001010 00000011*** 00000000 00000000
Free IS block	10.3.128.0/17	***00001010 00000011 1***0000000 00000000
Free IS block	10.3.64.0/18	***00001010 00000011 01***000000 00000000
Free IS block	10.3.32.0/19	***00001010 00000011 001***00000 00000000
Free IS block	10.3.16.0/20	***00001010 00000011 0001***0000 00000000
Free IS block	10.3.8.0/21	***00001010 00000011 00001***000 00000000
Free IS block	10.3.4.0/22	***00001010 00000011 000001***00 00000000
Free IS block	10.3.2.0/23	***00001010 00000011 0000001***0 00000000
Free IS block	10.3.1.0/24	***00001010 00000011 00000001*** 00000000
Free IS block	10.3.0.128/25	***00001010 00000011 00000000 1***0000000
Free IS block	10.3.0.64/26	***00001010 00000011 00000000 01***000000
Core Net IS block	**10.3.0.0/26**	***00001010 00000011 00000000 00***000000

3.1.3 Address Allocation Part 3

Now that we've allocated address space at the top level by application then at the core network level, each of these allocations can be subdivided further to serve requisite distribution center and branch office needs. In essence, these allocations serve as the block or pool of addresses that may be distributed for the given application within the respective region. This technique of top-down allocation ensures subsequent allocations from these initial allocations will roll-up hierarchically. Thus, our core routers can simply advertise their /16 allocations to the other core routers. Also, any special per service packet handling treatment can also be easily configured. For example, if we'd like to handle voice packets with highest priority treatment, we can configure our routers to provide such treatment for packets with source address from the respective voice space, such as 10.17.0.0/16 for Europe voice traffic (or 10.16.0.0/12 for all voice traffic). From this initial definition, further allocations can now be made further down geographical lines without affecting this treatment logic.

Let's drill into our North American data space, 10.32.0.0/16. From our location table presented earlier in Figure 3.1, we see that North American sites are organized in three regions: east, central, and west. We'd also like to allocate independent space for headquarters. Assuming our routing topology aligns with this geographical organization, we will allocate address space accordingly. Thus, a WAN may interconnect North American regional sites of Philadelphia, Kansas City, and San Francisco with headquarters. This regional interconnection represents a "subcore" network, and similar allocation logic can be applied as was at the top level.

Let's carve up our 10.32.0.0/16 block into four regional blocks. To allocate equally, we need to divide this space into four blocks. So we need to allocate the next 2 bits ($2^2 = 4$) in the North America data space, highlighted as larger font bold italic bits in the binary representation below.

N. America data	10.32.0.0/16	***00001010 00010000*** 00000000 00000000
N. America HQ data	10.32.0.0/18	***00001010 00010010 00***000000 00000000
N. America East data	10.32.64.0/18	***00001010 00010010 01***000000 00000000
N. America West data	10.32.128.0/18	***00001010 00010010 10***000000 00000000
N. America Central data	10.32.192.0/18	***00001010 00010010 11***000000 00000000

We don't necessarily have to allocate along powers of 2 as we're showing, though this results in equal-sized allocations. We could just as easily have allocated a larger portion to the east, since it contains the most sites: 10.32.0.0/17 (East), 10.32.160.0.0/19 (Central), 10.32.192.0/19 (West), and 10.32.220.0/19 (HQ).

From this point, we can allocate from each region's space to its respective sites for addressing needs. Considering the North America West data space, 10.32.128.0/18, we can now allocate space for data applications in each of our distribution centers and branch offices. The simplest strategy for such allocation is one of *uniform* distribution, for example, each site is allocated the same sized block as we performed at the top allocation level. However, one needs to consider the number of users and data devices per site, projected growth at each site, planned new sites within the region, and application networking requirements. In the case of IPAM Worldwide, distribution centers typically house 65 employees with additional automation machinery and infrastructure requiring IP addresses totaling around 200–250. Branch offices require only about 150–200 IP devices, including associate laptops, PDAs, and other data devices serving an average employee population of 40.

In such a scenario, it makes sense to allocate at least a /23 for distribution centers, providing 510 usable IP addresses, and /24 for branch offices, providing 254 IP addresses. However, each site should be analyzed individually regarding its respective addressing requirements. In our case, we'll first allocate a /23 per distribution center and then a /24 per branch office. The following table illustrates this allocation, along with the remaining free space from the original 10.32.128.0/18 network available for future allocation.

N. America West data	10.32.128.0/18	***00001010 00100000 10***000000 00000000
San Francisco 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

For our headquarters location, we'll allocate /22 networks for each of the major corporate divisions. These allocations may further be subnetted based on networking deployments.

3.1.4 Allocation Trade-Offs and Tracking

As you add layers in the address allocation hierarchy, the network portion of the address grows, shrinking the number of host bits assignable to IP devices. Each of the sites listed in the previous table has either 8 or 9 host bits available providing capacity for 254 or 510 individual IP hosts per site, respectively. Hierarchical layers enable mapping of address space to applications, regions, and ultimately, subnets, and help retain address summarization corresponding to router topology and deployments. It's a good idea to consider how much IP address capacity is needed at each site and trade this off with how many hierarchy layers are desired.

Individual IP address capacity requirements per subnet will help you derive the endpoint allocation size. Many organizations plan for allocating 254 hosts in a /24 allocation per end subnet. Multiple subnets could be allocated if needed. Using this octet boundary helps simplify translation from binary to decimal as you can see in the summary above, but it may not be feasible for your organization due to address capacity requirements. If you're required to allocate outside octet boundaries, use of an IP address management tool can probably help ensure accuracy of allocations without overlaps while conserving address hierarchy.

Whether you decide to use an IP management system or not, you must track address allocations. To illustrate one simple tracking method, we've recast our spreadsheet presented at the beginning of this chapter listing IPAM Worldwide's network locations to reflect respective block allocations. In the updated version shown below in Figure 3.5, we've listed distribution centers and branch offices together under a common Sites column, with distribution centers listed first in a lightly shaded font.

Our top-level hierarchical blocks that comprise the address supply at each hierarchy level are shown highlighted for each region to differentiate them from subnets. We followed a common allocation approach to keep things simple, allocating a /23 for each distribution center and a /24 for each branch office. We're only illustrating a small subset of the spreadsheet, but the same methodology is used for Europe and Asia sites and for voice and data applications.

A convenient side effect of this form of allocation yields the ability to easily associate an address with a location. For example, knowing that 10.0.79.0/24 is the infrastructure subnet for Albany, one could deduce that 10.16.79.0/24 is the VoIP subnet and 10.32.79.0/24 is the data subnet for Albany. This octet pattern of 10.X.Y.0 networks maps the application (octet X) and the location (octet Y) by sight. In our example, octet X is 0 for infrastructure, 16 for VoIP, and 32 for data. Octet Y is 79 for Albany in this example.

Region	Regional Site	Sites	Infrastructure Nets	VoIP Nets	Data Nets
HQ—Corp.	Phila.		**10.0.0.0/12**	**10.16.0.0/12**	**10.32.0.0/12**
HQ—N. Amer.	Phila.		**10.0.0.0/16**	**10.16.0.0/16**	**10.32.0.0/16**
		Core Net	10.3.0.0/26		
		Phila.—Exec	10.0.0.0/22	10.16.0.0/22	10.32.0.0/22
		Phila.—Fin.	10.0.4.0/22	10.16.4.0/22	10.32.4.0/22
		Phila.—Ops	10.0.8.0/22	10.16.8.0/22	10.32.8.0/22
		Phila.—Tech	10.0.12.0/22	10.16.12.0/22	10.32.12.0/22
		Phila.—Mktg	10.0.16.0/22	10.16.16.0/22	10.32.16.0/22
		Phila.—R&D	10.0.20.0/22	10.16.20.0/22	10.32.20.0/22
N. Amer—East	Norris—town		**10.0.64.0/18**	**10.16.64.0/18**	**10.32.64.0/18**
		Norristown	10.0.64.0/23	10.16.64.0/23	10.32.64.0/23

Figure 3.5. IPAM Worldwide's IPv4 block allocations (partial).

Region	Regional Site	Sites	Infrastructure Nets	VoIP Nets	Data Nets
		Toronto	10.0.66.0/23	10.16.66.0/23	10.32.66.0/23
		Nashua	10.0.68.0/23	10.16.68.0/23	10.32.68.0/23
		Newark	10.0.70.0/23	10.16.70.0/23	10.32.70.0/23
		Baltimore	10.0.72.0/23	10.16.72.0/23	10.32.72.0/23
		Pittsburgh	10.0.74.0/23	10.16.74.0/23	10.32.74.0/23
		Charlotte	10.0.76.0/23	10.16.76.0/23	10.32.76.0/23
		Atlanta	10.0.77.0/24	10.16.77.0/24	10.32.77.0/24
		Providence	10.0.78.0/24	10.16.78.0/24	10.32.78.0/24
		Quincy	10.0.79.0/24	10.16.79.0/24	10.32.79.0/24
		Albany	10.0.80.0/24	10.16.80.0/24	10.32.80.0/24
		Manhattan	10.0.81.0/24	10.16.81.0/24	10.32.81.0/24
		Ocean City	10.0.82.0/24	10.16.82.0/24	10.32.82.0/24
		Reston	10.0.83.0/24	10.16.83.0/24	10.32.83.0/24
		Richmond	10.0.84.0/24	10.16.84.0/24	10.32.84.0/24
		Charleston	10.0.85.0/24	10.16.85.0/24	10.32.85.0/24
		Montgomery	10.0.86.0/24	10.16.86.0/24	10.32.86.0/24
N. Amer.— Central	Kansas City		**10.0.192.0/18**	**10.16.192.0/18**	**10.32.192.0/18**
		Kansas City	10.0.192.0/23	10.16.192.0/23	10.32.192.0/23
		Chicago	10.0.194.0/23	10.16.194.0/23	10.32.194.0/23
		Des Moines	10.0.196.0/23	10.16.196.0/23	10.32.196.0/23
		Memphis	10.0.198.0/23	10.16.198.0/23	10.32.198.0/23
		New Orleans	10.0.200.0/23	10.16.200.0/23	10.32.200.0/23
		. . .	. . .	. . .	. . .

Figure 3.5. (Continued).

3.1.5 IPAM Worldwide's Public Address Space

Now let's look at IPAM Worldwide's public address space, 192.0.2.0/24, obtained from our ISP. We'll discuss the process ISPs use to get IP address space later in this chapter. IPAM Worldwide has an Internet connection to their chosen ISP from the Philadelphia headquarters office. While two diverse-routed local loops provide a level of access redundancy, future plans call for supporting a multihomed connection from another location, which we'll also discuss a bit later. For the time being, the 254 public IP addresses available within the /24 will be used to address Internet (externally) reachable hosts such as web and email servers, and a shared address pool to enable internal clients to access the Internet. A pair of NAT devices have been installed to enable load sharing and address translation for access by internal clients to the Internet. In reality, this /24 will likely need to be subnetted to partition Internet-reachable hosts from NAT addresses.

3.2 IPv6 ADDRESS ALLOCATION*

Though IPv6 addresses are represented differently than IPv4 addresses, the allocation process works essentially the same way. The main difference is in converting hexadecimal to binary and back instead of decimal to binary and back. The process of optimal assignment of the smallest available free block described above for IPv4 is an example of the best-fit allocation algorithm. Due to the vast difference in available address space, IPv6 supports not only an analogous best-fit algorithm but also a sparse allocation method. We'll also discuss a random allocation method that can be used in lieu of simple subnet numbering starting from 1 and counting up.

We'll outline each of these algorithms in this section, using the example IPv6 network 2001:DB8::/32. Note that /32 (or any)-sized global unicast allocations require prequalification with a Regional Internet Registry (RIR) as we'll discuss later in this chapter, and it's unlikely that an organization of the scale of IPAM Worldwide would receive such an allocation. However, we'll initially use this in our example to keep the number of bits from running off the page! Later we'll use a more practical /48 example allocation. The algorithm will be equivalent whether starting with a /32 or a /48, there'll just be more intermediate 0 prefix bits with the /48 network.

3.2.1 Best-Fit Allocation

Using a best-fit approach, we'll follow the same basic bit-wise allocation algorithm we used for IPv4 described earlier. After converting the hexadecimal to binary, the process is identical in terms of successive halving by seizing the next bit for the network portion of the address. For example, consider our example network 2001:0DB8::/32 below.

0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 0000 0000 0000...

* This discussion of IPv6 allocations is based on Ref. 172.

Let's say we'd like to allocate three /40 networks from this space. In following the analogous IPv4 allocation example from a binary perspective, by successively halving the address space down to a /40 size shown by the larger bold italic bits below, you should arrive at the following:

```
001000000000000100001101101110001000 0000000000000000...
00100000000000010000110110111000 0100 0000000000000000...
00100000000000010000110110111000 0010000000000000000...
00100000000000010000110110111000 000100000000000000000...
00100000000000010000110110111000 00001000000000000000...
00100000000000010000110110111000 0000010000000000000...
00100000000000010000110110111000 000000100000000000000...
0010 0000 0000 0001 0000 1101 1011 1000 0000 0001 0000 0000 0000...
0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 0000 0000 0000...
```

Here we readily have two /40 networks available (highlighted above), and translating these back into hex we have 2001:0DB8:0100::/40 and 2001:0DB8:0000::/40 (i.e., 2001: DB8::/40). After this allocation, to allocate a third /40 using the best-fit approach, we can then take the next smallest available network, in this case a /39, and split it into two /40s:

```
0010 0000 0000 0001 0000 1101 1011 1000 0000 0010 0000 0000 0000
0010 0000 0000 0001 0000 1101 1011 1000 0000 0011 0000 0000 0000
```

We split this into half by taking the next bit, yielding two /40s. We can choose one to allocate and the other will be free for future assignment. So our three /40s for allocation are 2001:DB8::/40, 2001:DB8:0100::/40, and 2001:DB8:0200::/40. The other /40, that is, 2001:DB8:0300/40, is available for future assignment. Figure 3.6 illustrates this successive halving in a pie chart form.

After allocating these three /40 networks, highlighted in Figure 3.6, the remainder of the pie is available for allocation. These available networks appear as the top six in the successive halving list above, plus the unallocated half of the former 2001:DB8:200::/39 network.

3.2.2 Sparse Allocation Method

You'll notice from the prior algorithm that by allocating a /40 from a /32, we incrementally extend the network length to the 40th bit as we did with IPv4 allocation. We then assign the network by assigning a 0 or 1 to the 40th bit as our first two /40 networks. In essence, we process each bit along the way, considering "1" the free block and "0" the allocated block. However, if we step back and consider the eight subnet ID bits that extend the /32 to a /40 as a whole, instead of incrementally halving the network, we observe that we've actually allocated our subnets by simply numbering or counting within the subnet ID field as denoted by the highlighted bold italic bits in this table:

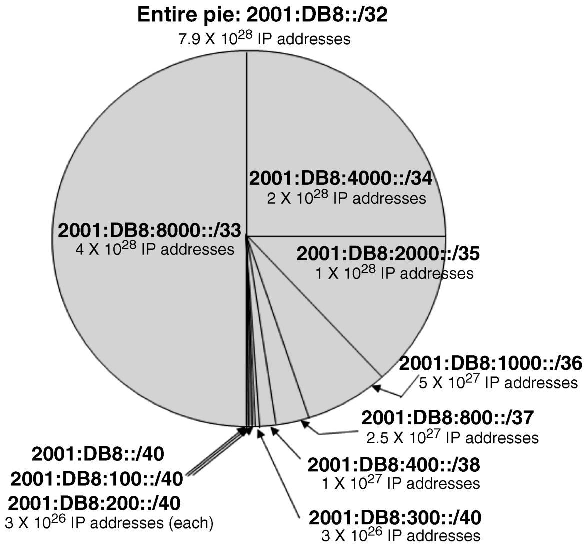

Figure 3.6. Allocation results from carving three /40 networks from a /32 network [Based on (11) and (166)].

0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 `0000 0000 0000... 2001:DB8::/40`
0010 0000 0000 0001 0000 1101 1011 1000 0000 0001 `0000 0000 0000... 2001:DB8:100::/40`
0010 0000 0000 0001 0000 1101 1011 1000 0000 0010 `0000 0000 0000...2001:DB8:200::/40`

Thus, if you knew in advance that the original /32 network would be carved uniformly into only /40-sized blocks, a simpler allocation method would be to simply increment the subnet ID bits. The next allocation of /40s would use subnet ID values of `00000011`, `00000100`, `00000101`, and so on. In some networks, this uniformity policy of allocating /40 blocks may not apply, so the method of successive halving may be more appropriate.

On the other hand, if you are a Local Internet Registry (LIR) or ISP, a sparse allocation method may be attractive. The sparse allocation method seeks to spread out allocations to provide room for growth by allocating with the maximum space *between* allocations. The sparse algorithm also features halving of the available address space, but instead of continuing this process down to the smallest size, it calls for allocating the next block on the edge of the new half. This results in allocations being spread out and not optimally allocated. Again, the philosophy is that this provides room for growth of allocated networks by leaving ample space between allocations in the plentiful IPv6 space. Considering an example, our allocation of three /40s from our 2001: DB8::/32 space would look like as below:

0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 0000 0000 0000... 2001:DB8::/40
0010 0000 0000 0001 0000 1101 1011 1000 1000 0000 0000 0000 0000... 2001:DB8:800::/40
0010 0000 0000 0001 0000 1101 1011 1000 0100 0000 0000 0000 0000...2001:DB8:4000::/40

These translate as 2001:DB8::/40, 2001:DB8:8000::/40, and 2001:DB8:4000::/40 respectively. This allocation enables spreading out of address space as illustrated in Figure 3.7. Should the recipient of the 2001:DB8:8000::/40 network require an additional allocation, we could allocate a contiguous or adjacent block, 2001:DB8:8100::/40. This block will be among the last to be allocated under the sparse method, so there's a good chance it will be available. In such a case, the recipient of our two contiguous blocks could identify (and advertise) their address space as 2001:DB8:8000::/39. Note that our subnet ID bits are effectively counted from left to right, instead of the conventional right-to-left method used for "normal" counting.

RFC 3531 (22) describes the sparse allocation methodology. Because network allocations are expected to follow a multilayered allocation hierarchy, several sets of successive network bits can be used by different entities for successive allocation. For example, an Internet Registry may allocate the first macro block to a Regional Registry, who in turn will allocate from that space to a service provider, who may in turn allocate from that subspace to customers, who can further allocate across their networks. RFC 3531 recommends the higher level allocations, for example, from the registries, utilize the leftmost counting or sparse allocation, the lowest level allocations use the rightmost or best-fit allocation, and others in the middle use either, or even a centermost allocation scheme. For an organization like IPAM Worldwide, we can use the sparse

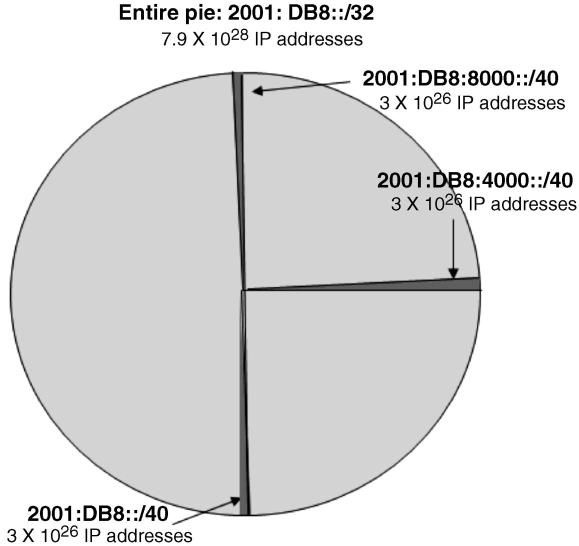

Figure 3.7. Sparse allocation example (Based on [11] and [166]).

method to allocate our intercontinental networks, leaving room for future growth at the top level. Note that while RFC 3531 addresses IPv6 allocation, we could also have allocated IPAM Worldwide's top-level IPv4 space in this manner to spread out initial allocations as 10.0.0.0/12 for infrastructure, 10.128.0.0/12 for VoIP, and 10.64.0.0/12 for data.

3.2.3 Random Allocation

The random allocation method selects a random number within the sizing of the subnetwork bits to allocate subnetworks. Using our /40 allocations from a /32, a random number would be generated between 0 and $2^8 - 1$ or 255 and allocated assuming it's still available. This method provides a means for randomly spreading allocations across allocated entities and generally works best for "same size" allocations. Randomization provides a level of "privacy" in not ordering blocks and subnets consecutively starting with "1." Be aware that random allocation may render the identification of larger contiguous blocks per our earlier merger example as well as the freeing up contiguous space for renumbering purposes more difficult. So while it makes sense to allocate sparsely at the top layer of allocation, the random or best-fit methods are more appropriate at the subnet allocation level.

3.2.4 Unique Local Address Space

While IPv6 does not have designated "private" address space, the concept of unique local address (ULA) space is essentially equivalent. By using the FC00::/7 prefix, setting the L bit to "1" (i.e., FD00::/8) indicating local assignment and assigning a random 40-bit Global ID, IPAM Worldwide can contrive a /48 network for internal use. There is some discussion among the Internet community over possibly enabling Regional Internet Registries, the organizations responsible for allocating public IP address space as we'll see next, to allocate globally unique 40-bit Global IDs. Regardless of global uniqueness, ULA destination addressed packets should not be routed outside an organization. Enforcement of this however is generally up to each organization to prohibit such packets from crossing beyond its external border routers.

3.3 IPAM WORLDWIDE'S IPv6 ALLOCATIONS

While we used a /32 for the sake of illustration when describing allocation, let's use a more realistic /48-sized block for the IPAM Worldwide example: 2001:DB8:4AF0::/48. And although IPAM Worldwide has plentiful public IPv6 space, for the sake of example, let's choose a ULA network for allocation as well: FD01:273E:90A::/48. These address blocks will be allocated hierarchically in accordance with IPAM Worldwide's geographic structure. We'll also allocate these networks using common subnet ID numbers for each location as in our earlier 10.X.Y.0 example, for pattern consistency and easier visual correlation.

In IPAM Worldwide's case, let's use sparse allocation at the core network layer. From this allocation, we can further sparsely allocate to our regions and then use a best-fit approach for our distribution centers and branch offices. While we're only running data applications over IPv6 for the time being, we should still perform an application-level allocation for application expansion or growth.

For simplicity, we will allocate on 4-bit boundaries.* If we use our first 4 subnet bits (bits 49–52), we have 16 possible allocations. Since we have one application as yet, we'll simply allocate four networks to represent our core networks for the "data" application. Using the sparse method, we arrive at the following allocations:

Core Allocation	Bits 49–52	Public Space Allocation	ULA Allocation
Headquarters	**0 0 0 0**	2001:DB8:4AF0::/52	FD01:273E:90A::/52
N. America	**1 0 0 0**	2001:DB8:4AF0:8000::/52	FD01:273E:90A:8000::/52
Europe	**0 1 0 0**	2001:DB8:4AF0:4000::/52	FD01:273E:90A:4000::/52
Asia	**1 1 0 0**	2001:DB8:4AF0:C000::/52	FD01:273E:90A:C000::/52

Applying a similar approach, using bits 53–56 for our next level allocations, we arrive at the following suballocations:

Subcore Allocation	Bits 53–56	Public Space Allocation	ULA Allocation
N. America—East	**0 0 0 0**	2001:DB8:4AF0:8000::/56	FD01:273E:90A::/56
N. America—Central	**1 0 0 0**	2001:DB8:4AF0:8800::/56	FD01:273E:90A:8800::/56
N. America—West	**0 1 0 0**	2001:DB8:4AF0:8400::/56	FD01:273E:90A:8400::/56
Europe—West	**0 0 0 0**	2001:DB8:4AF0:4000::/56	FD01:273E:90A:4000::/56
Europe—South	**1 0 0 0**	2001:DB8:4AF0:4800::/56	FD01:273E:90A:4800::/56
Europe—East	**0 1 0 0**	2001:DB8:4AF0:4400::/56	FD01:273E:90A:4400::/56

Within each of these /56 allocations, we can further allocate individual /64 subnet addresses for each distribution center and branch office. We'll perform this allocation using a best-fit approach and summarize a subset of our allocations in our expanded address allocation spreadsheet. IPv6 subnets in general should be allocated with /64 network prefixes. Many IPv6 features such as neighbor discovery assume (rely on) this prefix size.

For router point-to-point or back-to-back links, you may assign a /126 subnet, analogous to a /30 in IPv4 providing two host addresses. However, be aware of the setting of the "u" (universal/local, 71st bit) and "g" (individual/group, 72nd bit) bits within the interface identifier field of the IPv6 address. Setting these bits incorrectly may affect applications that access or utilize them. The "u" bit indicates that the company ID was assigned by the IEEE (1) or locally (0), and the "g" bit indicates that the address is a unicast (0) or a multicast (1). The /127 address should not be used. The /128 prefix denotes a single IP address, analogous to /32 in IPv4.

Let's add these IPv6 allocations to IPAM Worldwide's IP address spreadsheet that follows.

* We'll discuss the implications of allocating on non-4-bit boundaries, particularly on DNS, in Chapter 9.

Core Sites	Region	Regional Site	Sites	Infra. Nets	VoIP Nets	Data Nets	Public IPv6	IPv6 ULA
Philadelphia	HQ—Corp.	Philadelphia		**10.0.0.0/12**	**10.16.0.0/12**	**10.32.0.0/12**	**2001:DB8:4AF0::/52**	**FD01:273E:90A::/52**
Philadelphia	HQ—N. America	Philadelphia		**10.0.0.0/16**	**10.16.0.0/16**	**10.32.0.0/16**	**2001:DB8:4AF0::/56**	**FD01:273E:90A:8000::/52**
			Backbone Net	10.3.0.0/26			2001:DB8:4AF0:800::/64	FD01:273E:90A:800::/64
			Philadelphia—Exec	10.0.0.0/22	10.16.0.0/22	10.32.0.0/22	2001:DB8:4AF0::/64	FD01:273E:90A::/64
			Philadelphia—Finan	10.0.4.0/22	10.16.4.0/22	10.32.4.0/22	2001:DB8:4AF0:1::/64	FD01:273E:90A:1::/64
			Philadelphia—Ops	10.0.8.0/22	10.16.8.0/22	10.32.8.0/22	2001:DB8:4AF0:2::/64	FD01:273E:90A:2::/64
			Philadelphia—Tech	10.0.12.0/22	10.16.12.0/22	10.32.12.0/22	2001:DB8:4AF0:3::/64	FD01:273E:90A:3::/64
			Philadelphia—Mktg	10.0.16.0/22	10.16.16.0/22	10.32.16.0/22	2001:DB8:4AF0:4::/64	FD01:273E:90A:4::/64
			Philadelphia—R&D	10.0.20.0/22	10.16.20.0/22	10.32.20.0/22	2001:DB8:4AF0:5::/64	FD01:273E:90A:5::/64
	N. America—East	Norristown		**10.0.64.0/18**	**10.16.64.0/18**	**10.32.64.0/18**	**2001:DB8:4AF0:8000::/56**	**FD01:273E:90A:8000::/56**
			Norristown	10.0.64.0/23	10.16.64.0/23	10.32.64.0/23	2001:DB8:4AF0:8000::/64	FD01:273E:90A:8000::/64
			Toronto	10.0.66.0/23	10.16.66.0/23	10.32.66.0/23	2001:DB8:4AF0:8001::/64	FD01:273E:90A:8001::/64
			Nashua	10.0.68.0/23	10.16.68.0/23	10.32.68.0/23	2001:DB8:4AF0:8002::/64	FD01:273E:90A:8002::/64
			Newark	10.0.70.0/23	10.16.70.0/23	10.32.70.0/23	2001:DB8:4AF0:8003::/64	FD01:273E:90A:8003::/64
			Baltimore	10.0.72.0/23	10.16.72.0/23	10.32.72.0/23	2001:DB8:4AF0:8004::/64	FD01:273E:90A:8004::/64

(*continued*)

(*Continued*)

Core Sites	Region	Regional Site	Sites	Infra. Nets	VoIP Nets	Data Nets	Public IPv6	IPv6 ULA
			Pittsburgh	10.0.74.0/23	10.16.74.0/23	10.32.74.0/23	2001:DB8:4AF0:8005::/64	FD01:273E:90A:8005::/64
			Charlotte	10.0.76.0/23	10.16.76.0/23	10.32.76.0/23	2001:DB8:4AF0:8006::/64	FD01:273E:90A:8006::/64
			Atlanta	10.0.77.0/24	10.16.77.0/24	10.32.77.0/24	2001:DB8:4AF0:8007::/64	FD01:273E:90A:8007::/64
			Providence	10.0.78.0/24	10.16.78.0/24	10.32.78.0/24	2001:DB8:4AF0:8008::/64	FD01:273E:90A:8008::/64
			Quincy	10.0.79.0/24	10.16.79.0/24	10.32.79.0/24	2001:DB8:4AF0:8009::/64	FD01:273E:90A:8009::/64
			Albany	10.0.80.0/24	10.16.80.0/24	10.32.80.0/24	2001:DB8:4AF0:800A::/64	FD01:273E:90A:800A::/64
			Manhattan	10.0.81.0/24	10.16.81.0/24	10.32.81.0/24	2001:DB8:4AF0:800B::/64	FD01:273E:90A:800B::/64
			Ocean City	10.0.82.0/24	10.16.82.0/24	10.32.82.0/24	2001:DB8:4AF0:800C::/64	FD01:273E:90A:800C::/64
			Reston	10.0.83.0/24	10.16.83.0/24	10.32.83.0/24	2001:DB8:4AF0:800D::/64	FD01:273E:90A:800D::/64
			Richmond	10.0.84.0/24	10.16.84.0/24	10.32.84.0/24	2001:DB8:4AF0:800E::/64	FD01:273E:90A:800E::/64
			Charleston	10.0.85.0/24	10.16.85.0/24	10.32.85.0/24	2001:DB8:4AF0:800F::/64	FD01:273E:90A:800F::/64
			Montgomery	10.0.86.0/24	10.16.86.0/24	10.32.86.0/24	2001:DB8:4AF0:8010::/64	FD01:273E:90A:8010::/64
	N. America— KC Central			**10.0.192.0/18**	**10.16.192.0/18**	**10.32.192.0/18**	**2001:DB8:4AF0:8800::/56**	**FD01:273E:90A:8800::/56**
			Kansas City	10.0.192.0/23	10.16.192.0/23	10.32.192.0/23	2001:DB8:4AF0:8800::/64	FD01:273E:90A:8800::/64
			Chicago	10.0.194.0/23	10.16.194.0/23	10.32.194.0/23	2001:DB8:4AF0:8800::/64	FD01:273E:90A:8800::/64
			Des Moines	10.0.196.0/23	10.16.196.0/23	10.32.196.0/23	2001:DB8:4AF0:8801::/64	FD01:273E:90A:8801::/64
			Memphis	10.0.198.0/23	10.16.198.0/23	10.32.198.0/23	2001:DB8:4AF0:8802::/64	FD01:273E:90A:8802::/64
			New Orleans	10.0.200.0/23	10.16.200.0/23	10.32.200.0/23	2001:DB8:4AF0:8803::/64	FD01:273E:90A:8803::/64

We're going to need longer pages if this keeps up! As we noted with the IPv4 allocations where Pittsburgh, for example, uses "site number" 73 (third octet), we can identify its IPv6 site number as 8005, the fourth colon segment. We'll make one more allocation from our public IPv6 space for our externally (Internet) accessible servers, such as DNS, web, file transfer, and email servers. Our IPv4 space was allocated using two different address spaces: private space for internal allocations and public space for external. For IPv6, we've allocated public space and ULA space internally, and we need to add an allocation for external accessibility. Let's allocate the 2001:DB8:4AF0:2000::/56 network for assignment to external hosts.

Now we have completed our initial allocation planning for IPAM Worldwide and we have recorded each allocation in our spreadsheet. Let's take a step back and discuss how public IP address space is managed and allocated to ISPs and then describe multihoming (use of multiple ISPs) in more detail.

3.4 INTERNET REGISTRIES

IP addresses must be unique on a given network for proper routing and communication.* How is this uniqueness ensured across the global Internet? The Internet Assigned Numbers Authority (IANA) is responsible for global allocation of IP address space for both IPv4 and IPv6, as well as other parameters used within the TCP/IP protocol, such as application port numbers. In fact, you can view these top-level allocations by browsing to www.iana.org and selecting "Internet Protocol v4 Address Space" or "IPv6 Address Space" under Number Resources (23).

IANA is, in essence, the top-level Address Registry and it allocates address space to Regional Internet Registries (RIRs). The RIRs, listed below, are organizations responsible for allocation of address space within their respective global regions from their corresponding space allotments from IANA.

- AfriNIC (African Network Information Centre)—Africa region
- APNIC (Asia Pacific Network Information Centre)—Asia/Pacific region
- ARIN (American Registry for Internet Numbers)—North America region, including Puerto Rico and some Caribbean Islands
- LACNIC (Latin American and Caribbean Internet Addresses Registry)—Latin America and some Caribbean Islands
- RIPE NCC (Réseaux IP Européens)—Europe, the Middle East, and Central Asia

The goals of the RIR system are as follows:

- *Uniqueness*. Each IP address must be unique worldwide for global Internet routing.

* As with every seemingly authoritative statement, there are exceptions! Anycast addresses are typically assigned to multiple hosts, and multicast addresses likewise are shared. This statement applies to unicast addresses.

- *Aggregation*. Hierarchical allocation of address space ensures proper routing of IP traffic on the Internet. Without aggregation, routing tables become fragmented that could ultimately create tremendous bottlenecks within the Internet.
- *Conservation*. Not only for IPv4 but also for IPv6 space, address space needs to be distributed according to actual usage requirements.
- *Registration*. A publicly accessible registry of IP address assignments eliminates ambiguity and can help when troubleshooting. This registry is called the *whois* database. Today, there are many whois databases, operated not only by RIRs but also by LIRs/ISPs for their respective address spaces.
- *Fairness*. Unbiased address allocation based on true address needs and not long-term "plans."

As you can probably guess, the allocation methods we've discussed in this chapter are similar to those employed to allocate address blocks to RIRs, and by RIRs to allocate to National or Local Internet Registries, and in turn to service providers and end users. Allocation guidelines for RIRs are documented in RFC 2050 (24). The general address allocation hierarchy is depicted in Figure 3.8. National Internet Registries are akin to Local Internet Registries, but are organized at a national level.

Back in the 1980s and early 1990s, many corporations (end users per Figure 3.8) obtained address space directly from a centralized Internet Network Information Center (NIC). However, the RIR and LIR/ISP layers were inserted during the transition to CIDR addressing to provide further delegation of address allocation responsibility. Today, most organizations obtain address space from LIRs or ISPs. The process for obtaining such address space is generally dictated by the LIR/ISP with whom you conduct business, though RIRs recommend use of consistent policies to maximize efficiency.

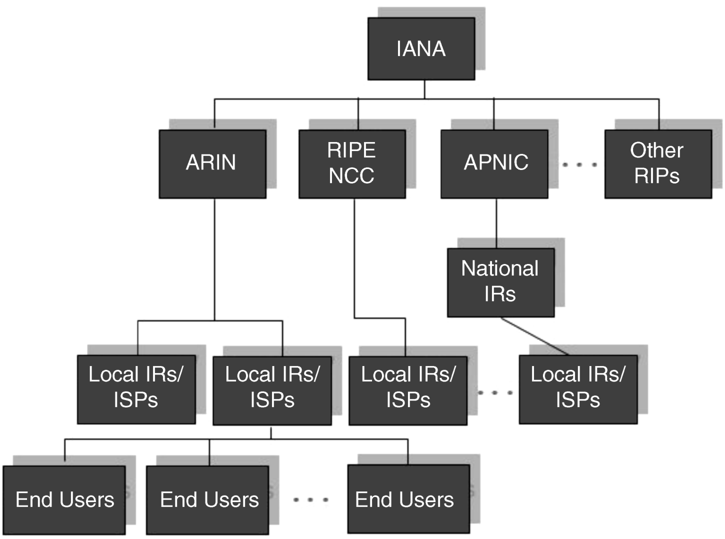

Figure 3.8. IP address allocation from the top-down (24).

As space is allocated to an ISP, the ISP may then advertise the address space on the Internet. Thus, this insertion of the LIR/ISP layer helps aggregate route advertisements on the Internet. Multiple customers served by the ISP can be summarized in one route on the Internet. If business is good and the LIR/ISP requires more address space, the LIR/ISP can request additional space from their RIR. Each RIR generally has its own defined process for fulfilling address requests, so you should consult the RIR in your region for details.

3.4.1 RIR address allocation

From an RIR perspective, RIRs *allocate* space to LIRs/ISPs, and LIRs/ISPs *assign* address space to their customers. The term *allocate* technically refers to the provision of an IP address block to serve as a "pool" of address space that can be drawn from for *assignment* to customers. Customers like IPAM Worldwide can then use the assigned address space, allocating blocks and subnets from it, and then assigning IP addresses from allocated subnets to individual hosts. The mechanics of this allocation and assignment are based on the procedures we described earlier in this chapter regarding the IPAM Worldwide hierarchical allocation examples. However, RIRs differentiate allocations from assignments because assignments comprise addresses in use, while allocations are pools for assignment that begin as unused but in theory grow in usage with a number of assignments from the allocation over time. Technically, RIRs count both allocations and assignments as in-use, but leave open the ability to audit allocated space for actual address utilization as needed to process additional allocation requests from each LIR/ISP.

To obtain address space in the first place, the LIR/ISP must demonstrate the need for utilization of 25% of the allocation immediately and 50% within 1 year*. This requirement applies to obtaining IPv4 space; it's a bit easier to obtain IPv6 space today. Requests for additional address space require justification via demonstration of utilization of the LIR/ISP's current allocations. In order to keep track of LIR/ISP allocations, the RIRs have each implemented electronic update mechanisms. As the LIR/ISP assigns address space, the assignment information can be communicated to the RIR using the corresponding form of electronic update. Theoretically, by the time additional address space is requested, the RIR and LIR/ISP have common allocation information against which the utilization threshold can be confirmed and approved. All RIRs allow emailing of information in specific template formats to convey allocation information. ARIN refers to this process as the Shared Whois Process or SWIP. Email templates vary by RIR and do occasionally change, so it's best to contact the RIR serving your geographic region to obtain the latest version of the template you require.

To control allocation of diminishing IPv4 address space, RIRs employ a *slow-start* allocation scheme to enable LIR/ISPs to start with a small address allocation and obtain larger allocations as their allocation performance dictates. In some instances, the LIR/ISP must obtain approval from the RIR prior to allocating from its space to an end user. RIPE, APNIC, and LACNIC RIRs utilize a construct called the *assignment window*

* Note that with rapidly diminishing availability of IPv4 space, allocation policies are rapidly evolving and tightening. Please contact your RIR for the latest allocation policies.

(AW) to control the block size allocations requiring RIR approval; AfriNIC uses a similar construct called a suballocation window (SAW). ARIN utilizes a standard /20 as its effective AW. The AW or SAW, expressed in CIDR notation, enforces boundaries constraining the maximum size of block allocations made by LIR/ISPs from their address space without approval from the corresponding RIR.

While APNIC and LACNIC enforce the AW (and AfriNIC's SAW) as the maximum allocation size without RIR approval, RIPE enables an LIR/ISP to allocate up to 400% of their AW or up to a /20 whichever is smaller. Thus, an LIR/ISP with an AW of /22 from RIPE may make individual allocations of up to /20 (remember, each bit movement to the left doubles the address space), while one with an AW of /21 may make individual allocations of up to a /19. AfriNIC, RIPE, LACNIC, and APNIC constrain the [S]AW to zero for new LIR/ISPs, requiring RIR approval of all allocations. Over time, as the LIR/ISP proves a responsible steward of address space, the [S]AW can be increased. Table 3.1 summarizes the key allocation policies employed by the RIRs today.

To enforce the aggregation goal of the RIR system, the RIR strongly recommends that each LIR/ISP contractually oblige their customers to return address space allocated to them should they decide to change service providers. This goal exists to preserve the addressing hierarchy, analogous to the addressing hierarchy we described in IPAM Worldwide's case. To illustrate the importance of this requirement, consider an ISP with a /20 allocation from a RIR. The ISP allocates a /23 to one of its customers. If the

TABLE 3.1. RIR allocation policy summary

RIR Policy Highlights	Regional Internet Registry				
	AfriNIC (25)	APNIC (26)	ARIN (27)	LACNIC (28)	RIPE NCC (29)
Initial minimum IPv4 allocation	/22	/21	/20	/21	/21
IPv4 justification: initial/1 year	25/50%	25/50%	25/50%	25/50%	25/50%
IPv4 network utilization criteria for additional allocations	80%	80%	80%	80%	80%
RIR must be notified for allocations greater than:	Assigned SAW	Assigned AW (/19 max)	/19 (or /18 for extra large ISPs)	Assigned AW (/21 max)	/20 or 4 X AW
Initial minimal IPv6 allocation	/32	/32	/32	/32	/32
Expected LIR/ISP IPv6 assignment	/48	/48	/48	/48	/48
IPv6 HD ratio criteria for additional allocations	0.94	0.94	0.94	0.94	0.94

customer changes service providers and takes the /23 address space with them, the /23 will need to roll up on the Internet "backbone" to the new ISP, not the original ISP. The original ISP would now need to advertise not only the /20 block it had originally received but also the other /23, a /22, and a /21; that is all of its space minus the departed customer's /23. The new ISP would need to advertise its allocated blocks plus the new customer's /23. This obviously creates a rise in routing table entries on both sides and on the global Internet, defeating the goal of aggregation. Such "portable" address space is referred to as *provider independent* (PI) and is easy for customers but inefficient for Internet routing.

By requiring a return of the address space to the original ISP, the /23 can be returned to the pool to be assigned or suballocated elsewhere, and the customer will need to renumber their IP network with new address space assigned by the new ISP. The requirement that address space be returned upon changing ISPs ensures the ISP a single aggregate route advertisement and is termed *provider aggregate* (PA) space. Some organizations that obtained IP address space prior to the institution of the ISP/LIR layer in the allocation hierarchy have PI space. Now allocation of PI space is largely frowned upon in favor of PA space.

3.4.2 Address Allocation Efficiency

During the development of IPv6, much thought went into deriving the 128-bit address size. While IPv4 provides a 32-bit address field that provides a theoretical maximum of 2^{32} addresses or over 4.2 billion addresses, in reality the theoretical maximum is much less than 4.2 billion. This is due to the hierarchical allocation of address space for multiple layers of networks, then subnets, and finally hosts. RFC 1715 (Ref. 30) provides an analysis of address assignment efficiency, in which a logarithmic scale was proposed as a measure of allocation efficiency, which was defined as the H ratio:

$$H = \frac{\log_{10}(\text{number of objects})}{\text{number of available bits}}$$

With about 730 million hosts on the Internet today, today's H ratio is 0.277. The H ratio for 100% utilization of 4.2 billion IP addresses is 0.301,* so today's H ratio is high.

Assignment efficiency measurements for IPv6, with its massive amount of address space, is not calculated based on the H ratio; a different ratio, the HD ratio (31), is used:

$$HD = \frac{\log_{10}(\text{number of allocated objects})}{\log_{10}(\text{maximum number of allocatable objects})}$$

The "objects" measured in the HD ratio for IPv6 are the IPv6 site addresses (/48s) assigned from an IPv6 prefix of a given size. The /48 address blocks are those expected to be assigned to each end user by the LIR/ISP. So an LIR/ISP with a /32 allocation that has allocated 100 /48s would have an HD ratio of log(100)/log (65,536) = 0.415.

* 0.301, which is equal to $\log_{10}(2)$, is the maximum value of the H ratio.

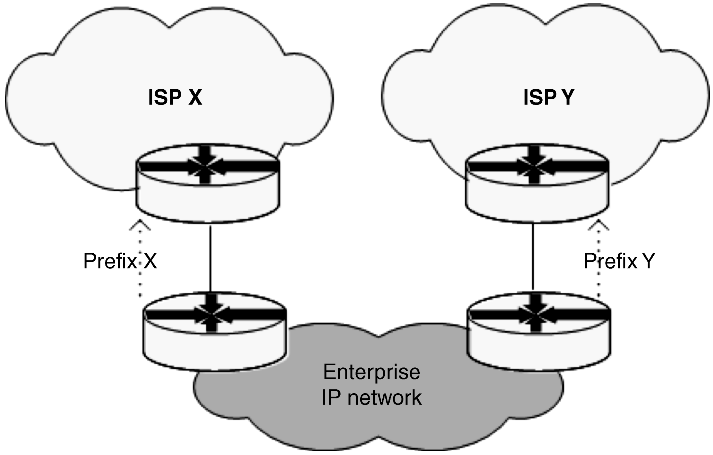

Figure 3.9. Multihoming architecture.

3.5 MULTIHOMING AND IP ADDRESS SPACE

The term *multihoming* refers to an enterprise provisioning multiple (>1) connections to the Internet. A simple architecture is depicted in Figure 3.9. A multihoming strategy provides several benefits:

- Link redundancy, providing continued Internet connectivity availability in the event of a connection outage
- ISP redundancy if multiple ISPs are used to limit exposure in the event of an ISP outage
- Load sharing of Internet traffic over multiple connections
- Policy and performance benefits achieved through routing of traffic based on congestion or based on requirements to route traffic of differing applications to differing links or ISPs.

Multihoming offers several attractive benefits though it does require care in configuring routers interfacing to each ISP. As we show in Figure 3.9, the enterprise border routers interfacing directly to their respective ISP edge routers participate in an exterior routing protocol (e.g., BGP) to advertise reachability to the respective address blocks (by address prefix). Thus, the enterprise router connected to ISP X will advertise reachability to the address space provided to the enterprise by ISP X. Similarly, the enterprise router connected to ISP Y will advertise reachability to the address space provided by ISP Y.

These two enterprise routers also communicate with each other using an interior routing protocol via the enterprise IP network. In this manner, loss of connectivity to an ISP may be detected, though this is where things get interesting. To illustrate this without going into all the routing details,* the following summarizes the most common multihoming deployment options, outage impacts, and implications on IP address space:

- *Case 1.* Two or more diverse physical links to the same ISP. This "multiattached" architecture provides link redundancy but not ISP redundancy. Referring to Figure 3.9, the two ISP clouds would be collapsed into a single cloud but still with two (or more) links from the enterprise. With one ISP, prefix X = prefix Y, so this public address space allocated from the ISP may be advertised uniformly on all connections.
- *Case 2.* Two or more connections to one or more ISPs using provider independent address space. Recall that PI space in this scenario has been allocated to the organization directly and independent of ISP associations. Referring to Figure 3.9, the advertised prefix is again the same on both connections, though we could denote it as prefix Z as being independent of the ISP address space. As in case 1, the PI space may be advertised to all ISPs and allocated across the organization as needed.
- *Case 3.* Two or more connections to two or more ISPs using provider aggregate address space from each ISP. In this case, each ISP allocates address space as part of its service. Figure 3.9 reflects this scenario as is. With two independent address blocks X and Y, if the link to ISP X fails, the enterprise router connected to ISP Y will detect this by virtue of the interior routing protocol update from the enterprise router connected to ISP X. Thus, the enterprise router connected to ISP Y could now advertise reachability to prefix X. Depending on ISP Y's policies, it may or may not propagate the route because it does not fall within ISP Y's address space but ISP X's.

Another approach is to perform an indirect BGP peer update from the ISP Y-facing enterprise router to an ISP X router peer. In this manner, ISP X routers may be notified of an alternate route to reach the enterprise's address space via ISP Y. These two alternative approaches are shown in Figure 3.10 with the former shown with prefix X being advertised to ISP Y's router and the latter with prefix X being advertised to ISP X's router.

NAT gateways at each ISP connection are commonly deployed to enable address pooling, translating a given packet's internal private address into a public address based on the ISP connection proximity, for example, from prefix X or prefix Y. Barring the use of NAT gateways, enterprise border router policies should be implemented to minimize or prevent routing of traffic among internally addressed hosts via the ISP(s).

3.6 BLOCK ALLOCATION AND IP ADDRESS MANAGEMENT

In this chapter, we've discussed techniques for IP address block allocations for public and private IPv4 space and IPv6 space. From a basic management perspective, it's critical to

* Please refer to the following RFCs for the routing details around multihoming: 2260 (170), 4116 (171), 4177 (172), and 4218 (173).

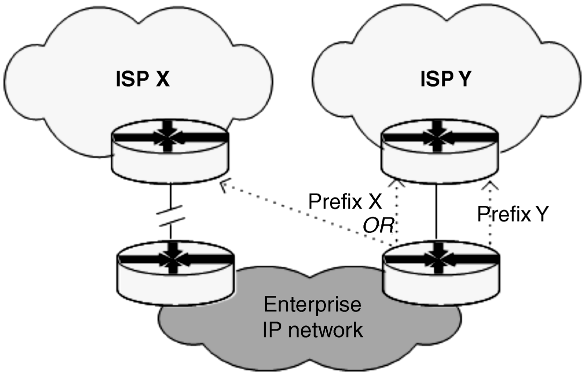

Figure 3.10. Multihoming link outage recovery.

maintain an inventory of allocations with respect to the hierarchy of allocations, in which locations of subnets have been deployed, and any additional relevant information such as a mapping of each subnet to its corresponding provisioned router or switch interface, associated Internet Registry administrative information if appropriate, trouble contact information, and other information of interest.

Many organizations responsible for block allocations use spreadsheets or simple database applications such as Microsoft Access to keep this information organized. Though performing block allocations and binary arithmetic are not natively performed with a single mouse click using spreadsheets, underlying homegrown Visual Basic or perl code, for example, can apply the logic we've reviewed in this chapter in order to perform best-fit, sparse, or random allocations and track the resulting allocations. Certainly, care must be taken to perform this accurately and to manage block allocations, moves, modifications, and deletions.

We've kept the IPAM Worldwide spreadsheet updated with each allocation to reflect our private IPv4, public IPv4 and IPv6, and ULA IPv6 address allocations. This information provides a necessary foundation for our next step, assigning individual IP addresses from respective subnets for each location. Now that each branch office and distribution center has IP subnets that aggregate up the hierarchy, we can begin the process of address assignment. We'll discuss one form of automated address assignment, DHCP, in the next part of the book, and we'll discuss other more manual methods of address assignment in Chapter 14.