Essential IPAM

OSI, Addressing, Routing, and More

VOLUME 3

Introduction to IP Routing

This paper examines the basics of routing protocols and the logic behind various routing methods. It is the third part of the Essential IPAM series. It is meant to be an introductory level paper.

Table of Contents

SECTIoN 1

Introduction

In this volume, we will examine how data is routed in Layer 3 and the logic used to make routing decisions. You should have a basic understanding of Variable Length Subnet Mask (VLSM) and classless IP addressing to get the most out of this material. If you are not familiar with those technologies, Essential IPAM Volume 2, IP Addressing is suggested.

SECTIoN 2

Routing Concepts

For the purpose of this paper, I'll define routing as the process of moving data from one IP network or subnet to another IP network or subnet. A router has two responsibilities in this process. First keep an updated database of routing for the networks the router is aware of, and secondly, when a packet is received for a known network, send that packet to the next hop or local segment for delivery. Obviously the first must be complete for the second to be possible. Otherwise routing network traffic would be like driving on unmarked roads with no signage; plenty of interconnecting highways but no way to tell which road goes where.

In general, **routes** can be categorized as:

1. **Learned routes** — learned by receiving routing information from other routers and applying that information to the routing protocol.
2. **Local segments** — The router knows these because the network or subnet is directly connected to the router and configured for IP in the router configuration.
3. **Static routes** — routes that have been defined by a network administrator.

Routing protocols can be broken in to two high level categories:

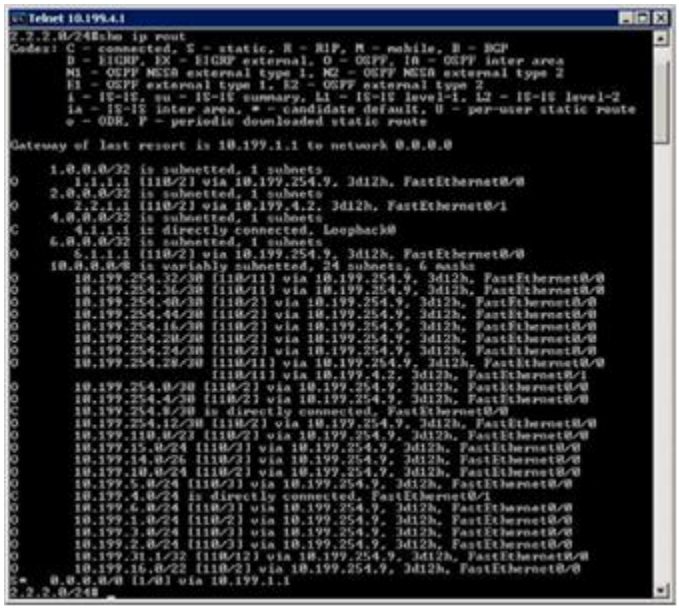
1. Interior routing
2. Inter-domain routing

The individual routing protocols can be further categorized as the algorithms they use to determine routes. These are:

1. **Distance-vector**. These protocols use simple costs of possible paths to determine the best route. Cost, for these protocols, is usually a function of the number of hops required to reach a target network.
2. **Link-state**. In these protocols, routers exchange information about their connections and each router then creates a logical map of the network from that router's perspective. The router can then make routing decisions with knowledge of the whole network and route costs for each segment based on data such as bandwidth and hop count.
3. **Path-vector**. These protocols use high-level path information to advertise the reachability of networks. The path information may include rolled-up routing information such as network x is reachable across the path AS a - AS b - AS c. This is implemented extensively in Internet routing using **Border Gateway Protocol (BGP).**

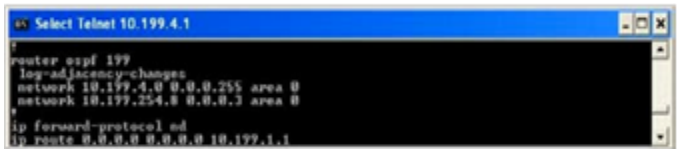
Below (see **Figure 1**) is the output of a show IP route command on a Cisco router.

Figure 1.



Each of the three types of routes can be seen in this capture. All of the routes marked with an "O" in the left margin are learned, Open Shortest Path First (OSPF) routes. The two marked "C" are directly connected segments. The last route marked "S" is a **static route** to the default gateway, sometimes called the **route of last resort** or **gateway of last resort**. The connected interfaces are defined by the interface configuration in the router's running configuration. OSPF routing is enabled as shown below to allow this router to be part of the **Autonomous System** (AS) area involved. The configuration snippet is shown below in **Figure 2**.

Figure 2.



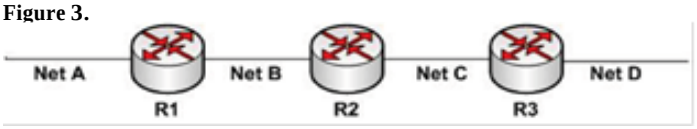
The static, default route is defined in the configuration, but the default route could also be learned from a route advertisement sent from another router

SECTION 3

Routing Mechanisms

Probably the best way to understand routing and the mechanism employed is to examine simple networks and issues with routing. Once we have a basic understanding of these mechanisms, we will explore individual routing protocols and see how they are implemented. Consider the simple network representation below (**Figure 3**).

Figure 3.



The routers on this network will participate in routing by sharing routing information in packets called **route advertisements**, which are broadcast out to all directly connected segments. For now, these simple advertisements will just inform other routers of what networks each router can get to and how many hops away the networks are. This is distance-vector routing information. The distance is the number of hops and the vector (direction) is the router from which the advertisement originated. When R2 is powered up, the only networks the router will know about are the segments connected to R2. R2's route table would look something like this:

```
R2#
Net B is directly connected
Net C is directly connected
```

Now when R1 and R3 share their known routes, the directly connected segments for each, R2 will add these to its table, which will now look like this.

```
R2#
Net B is directly connected
Net C is directly connected
Net A is reachable via R1 1 hop away
Net D is reachable via R3 1 hop away
```

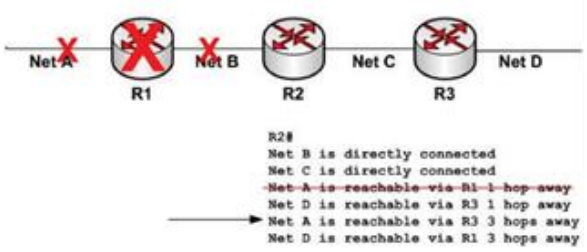
So far so good. R2 now knows how to get to all 4 networks. After some more route advertisements, R3 and R1 know how to reach the networks on the far side of the networks, Net A for R3 and Net D for R1. So R1 would advertise to R2 that R1 has a route to Net D at 2 hops. The same would happen for the route advertisements R3 sends to R2. This advertisement would tell R2 that Net A is reachable through R3 at a cost of 2 hops. Because these routers are only keeping distance-vector information they have no concept of the network topology. Here is what R2's route table now looks like.

```
R2#
Net B is directly connected
Net C is directly connected
Net A is reachable via R1 1 hop away
Net D is reachable via R3 1 hop away
Net A is reachable via R3 3 hops away
Net D is reachable via R1 3 hops away
```

We have not yet made any rules about how a router handles multiple routes to a network, for now, the routers just add them to the table. If this continues infinitely, then R2 will have an infinite number of routes to Net A and Net D as the routes are advertised back and forth. This is called the **count to infinity** problem. Here we'll introduce our first rule, a **maximum hop count** rule; only advertise routes that are 15 hops or less away. While this may seem like too large of a number for this network, it allows for larger networks with many routers and many hops between them. So now R2 will eventually have 15 hop routes to networks A and D which are not really useful, but at least we have solved the count to infinity issue.

Now let's say that R1 fails as seen below **Figure 4**.

Figure 4.



After a while R2 will receive no more route updates from R1 so it will age out the routes from R1 and eliminate them from R2's routing table. Here we have introduced an **age out timer**. If R2 has to route packets to Net A, R2 now believes that Net A is reachable by the next available path with the lowest hop count. This is the 3 hop route via R3 as seen above. Because these routers do not share topology information, R2 does not know that Net A is not really available at all. We are going to need a new rule to deal with this issue.

The root of this issue was that R2 was advertising routes back to R1 that it had learned from R1. R2 was also doing the same for routes it learned from R3. So here is the rule to solve this; don't advertise any routes back to where you learned them. This is known as **split horizon route advertising**. Putting split horizon rules into place, R2's route table would look like this before the R1 failure was discovered.

```
R2#
Net B is directly connected
Net C is directly connected
Net A is reachable via R1 1 hop away
Net D is reachable via R3 1 hop away
```

And it would look like the below after R2 aged out the routes from R1.When all changes in routing tables become stable throughout the network, we have reached what is called **route convergence**.

```
R2#
Net B is directly connected and down
Net C is directly connected
Net D is reachable via R3 1 hop away
Net A is unreachable
```

R2 will now drop any packets it receives for delivery to Net A. For a more complex network, it may not be enough to just split the horizon by sending no advertisement back to where they came from. We can implement another rule that keeps routes from bouncing back and forth. Now what we will do is when R2 learns a route from R1, R2 returns that route to R1 but marks it unreachable via R2. This is called a **poison reverse route advertisement**. The result is the same as split horizon, but this is a more active solution.

Another issue that may occur on this simple network is that the local loop on R1 that attaches to Net B could have intermittent problems. To R2 this might look like R1 and Net A are available, then unavailable, then available. This issue is called a **flapping interface**. Left unchecked, this could cause R1 and Net A to flap back and forth in R2's routing table from available to unavailable. If R2 then forwards all of these changes as they happen, the result could be a network flooded with route updates. One way to solve this issue is to place a timer on the routers so that when a router sees a network as unreachable, it doesn't advertise this to other routers for 90 seconds. This is called a **hold down timer**.

SECTION 4

Building an RIP Routed Network

What we have created is a routing protocol very similar to **Routing Information Protocol Version 1(RIPv1).** Now we'll take this protocol and apply it to a simple network. Below is our network (**Figure 5**).

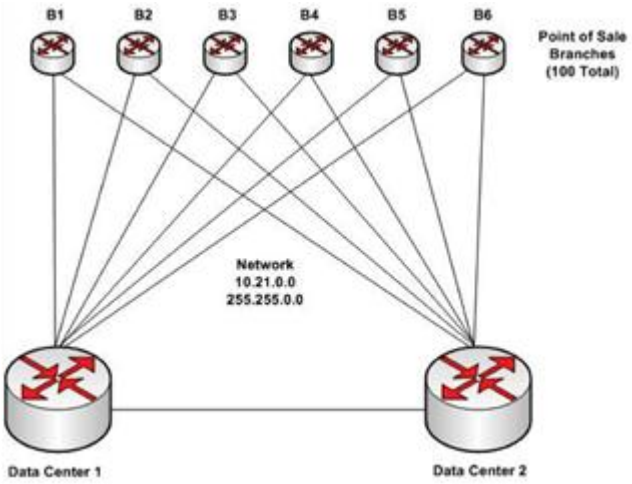


Figure 5.

In addition to what is shown here, the data centers have several routers connecting to business partners, the Internet, and VLANS within each data center. The Point of Sale (POS) branches need to be able to access data center servers to process complicated transactions and they need to have as much of the configured bandwidth available to them as possible. We will use RIPv1, which has all routers advertising their routes every 30 seconds. With all of the data center routers and all the branch routers combined, the routing tables could be quite large. This might consume significant bandwidth to the branch routers, especially seeing as they are all dual attached. To load balance, 50 of the 100 branches should connect to data center 1 as a primary connection and use the data center 2 connection as a backup connection. The other 50 should use the opposite configuration.

One solution would be to configure a default route of 0.0.0.0 at each branch with the primary data center and not advertise any routes for the data centers to the branches. When a router has only one default route configured, the router sends all packets for unknown networks to that route. So this will work for the connection back to the primary data center but leaves the branch without a route to the secondary data center. Seeing as a router in this routing protocol can only have one default route, this will not allow for dynamic, dual attached POS branches. Earlier I mentioned that default routes can be learned from route advertisements. We can take advantage of this to dynamically set default routes for the branches. Here is the routing design:

- On data center 1's router, do the following.
  - Do not forward any learned routes from the core network out to the branches.
  - For all odd number branches, send out a default route of 0.0.0.0 with the next hop of data center 1 router with a route cost statically set at 2.
  - For all even number branches, send out a default route of 0.0.0.0 with the next hop of data center 1 router with a route cost statically set at 4.
- On data center 2's router, do the following.
- Do not forward any learned routes from the core network out to the branches.
- For all even number branches, send out a default route of 0.0.0.0 with the next hop of data center 2 router with a route cost statically set at 2.
- For all odd number branches, send out a default route of 0.0.0.0 with the next hop of data center 2 router with a route cost statically set at 4.
- On all branch routers do the following.
- Configure to listen for RIP advertisements.
- Configure to send RIP advertisements.

RIP can only have one default route at any time, which causes the branch routers to keep the lowest cost route. So all the even number branches will keep a route to data center 2 and all the odd number ones will keep a route to data center 1. Because these routes are advertised from the data centers, if one data center fails, all of the routers that used that data center as primary will stop getting updates from that data center and age out that route. Now the best route these branches will have is the 4 hop route to the secondary data center. The only advertisements the branches receive are 2 route entry updates from each data center. The only advertisements the branches send to the data centers is for the directly connected branch LAN. Seeing as these are very small updates, the consumed bandwidth is very low. This type of routing trick can be used in other routing protocols as well.

RIPv1 limitations

RIPv1 was widely used from 1988, when it was first introduced, to the mid-90s. Before there was a RIPv2 it was just known as RIP. While RIPv1 is good for demonstrating some of the concepts and mechanisms used in routing, it is for the most part obsolete today. As internetworking became widely implemented in the build-out era of the 90s, several limitations of RIPv1 led to its demise. These include:

- 15-hop maximum routing distance.
- Network traffic overhead due to route advertisements broadcast at regular intervals, usually every 30 seconds.
- RIP1 does not allow for routing areas.
- Variable Length Subnet Masks (VLSMs) and Classless Internet Domain Routing (CIDR) are not supported.

The last point alone is enough to doom RIP1 as CIDR and VLSM are now almost universal. RIP can work well in Small Office/Home Office (SOHO) LAN networks because of its simplicity and the lack of a need for SOHO networks to route, other than a single route out the gateway to the service provider.

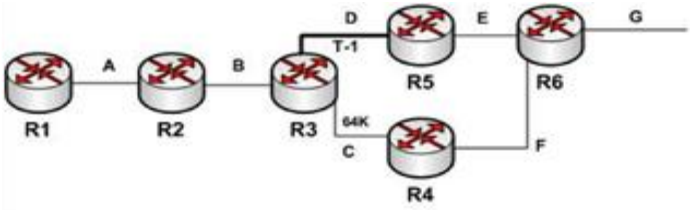
RIPv2 was created to address some of these issues, mostly to add the support for CIDR/VLSM. RIPv2 remains very limited in function compared to other types of routing, namely link-state routing protocols such as **Open Shortest Path First (OSPF)** and **Intermediate System to Intermediate System (IS-IS).** OSPF has become the interior routing protocol of choice. In fact, most routing today is done by two routing protocols, OSPF for interior routing and BGP for inter-domain routing.

SECTION 5

OSPF Routing

OSPF is a link-state routing protocol. While the main job of routing is always to get the data to the next hop as quickly as possible, how the routers understand the best next hop is a critical difference. In distance-vector routing, the routers know nothing about the topology of the network. All they know is that networks are available at a certain interface and are a certain distance away. Take the below network for example (see **Figure 6**):

Figure 6.



If we were to implement a distance-vector routing protocol, R1's routing table would show segment G available via R2 at 4 hops. This is regardless of the path chosen by R3, the 64K segment C or the T-1 segment D. R3 would make this decision only based upon distance, so segment D and segment C appear the same to R3. R3 would only make the decision on which route to keep based on the first route it discovered. There is a 50-50 chance the router would keep segment with only 64K bandwidth. R3 would then advertise this to R2, which would advertise it to R1.

Now if we implement OSPF, the link-state database will allow a much better decision making process. The primary benefit of a link-state routing protocol is that each router builds and uses a logical map of the entire routing domain, the AS. Instead of using a simple, raw metric such as hop count, OSPF routers assign a **path cost** value to routes according to the bandwidth of the entire path. OSPF calculates this cost for each segment by dividing a **reference bandwidth** by the circuit's configured bandwidth. Cisco uses a default reference bandwidth of 100Mbps. So a 100 Mbps circuit has an OSPF cost of 1 (100/100). A T-1 segment would have a cost of 100/1.544 = 65. This is actually rounded up from the more exact answer of 64.77. OSPF costs are represented by positive integers only with a minimum of 1 and a maximum of 65535.

With this reference, bandwidth with any segment faster than 100 Mbps will receive a cost of 1, but low speed circuits such a 512Kbps and 64 Kbps will be differentiated. This is shown below in **Table 1**.

Table 1.

Ref Mpbs	100000000	
Segment Speed (bps)	Common Name	Cost
100,000,000,000	100 Gig	1
10,000,000,000	10 Gig	1
1,000,000,000	1 Gig	1
100,000,000	100 Meg	1
45,000,000	T-3	2
10,000,000	10 Meg	10
1,544,000	T-1	65
512,000	512	195
256,000	256	391
64,000	64K	1563

The problem here is easy to see. If we care about the costs for any circuits greater than 100 Meg, we will have to adjust the reference bandwidth. Below, in **Table 2**, are the same speed segments with the OSPF cost calculated against a 100 Gig reference speed.

Table 2.

Ref Mpbs	100000000000	
Segment Speed (bps)	Common Name	Cost
100,000,000,000	100 Gig	1
10,000,000,000	10 Gig	10
1,000,000,000	1 Gig	100
100,000,000	100 Meg	1000
45,000,000	T-3	2222
10,000,000	10 Meg	10000
1,544,000	T-1	64767
512,000	512	65535
256,000	256	65535
64,000	64K	65535

What we have demonstrated is that adjusting the reference speed upward allows for greater differentiation on faster segment at the cost of less differentiation for lower speed segments. One work around is that you can manually set the OSPF cost on individual segments to allow for differentiation. If we were to use the reference of 100 Gig, we could possibly set the low speed costs as seen below in **Table 3**.

Table 3.

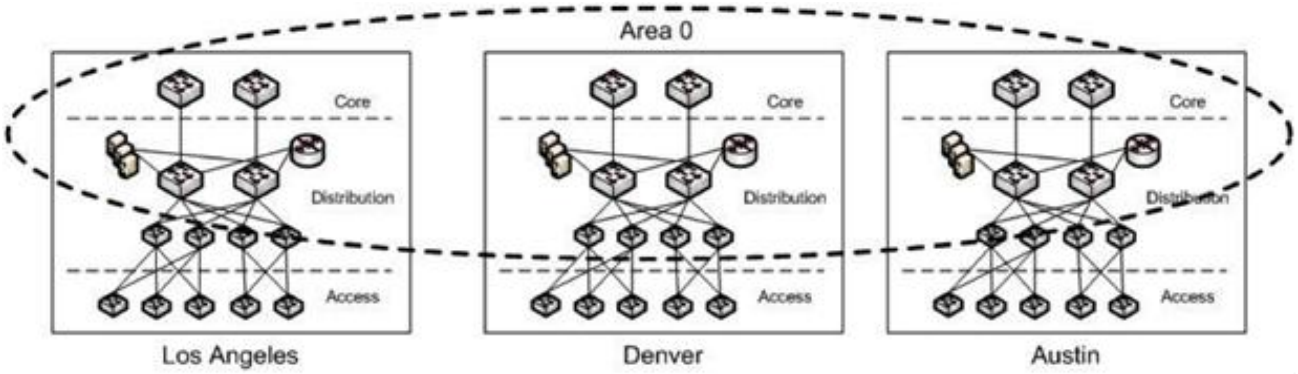
Ref Mpbs	100000000000	
Segment Speed (bps)	Common Name	Cost
100,000,000,000	100 Gig	1
10,000,000,000	10 Gig	10
1,000,000,000	1 Gig	100
100,000,000	100 Meg	1000
45,000,000	T-3	2222
10,000,000	10 Meg	10000
1,544,000	T-1	35000
512,000	512	45000
256,000	256	55000
64,000	64K	65000

While higher speed circuits will route correctly because we have them set to the OSPF calculated cost. What we do not know is how paths containing low speed segments will behave. Because OSPF routers know the entire network topology in their reference maps, they pick the lowest cost end-to-end path. I think the best way to determine how routes will work in a production network is to model them out as much as possible. A great tool for doing this is Dynamips or the GNS3 graphical Dynamips interface. This software runs actual Cisco IOS and Juniper JOS on a simulated network you create. It is like drawing your network in Visio and turning the routers and switches on to see how it all will operate. It is a cool way to learn Cisco and Juniper networking. The platform choices are somewhat limited but because you are running actual router OS you can manage that virtual network with Network Performance Monitor, NetFlow Traffic Analysis, and IP SLA Manager!

To share routing data, OSPF routers first need to understand what routers they are directly connected to (**adjacent** devices). The routers send out "Hello" packets to determine adjacencies and to then work out with the other routers a neighbor number for each router, a **Designated Router (DR) interface**, and a **Back-up Designated Router (BDR) interface**. There is a complicated procedure the routers follow to determine the DR but it usually boils down to the router with the highest IP address on an active interface. The DR interface is an important item in OSPF as it designates the handling of broadcast and multicast traffic across the AS. You may not want to leave this assignment to the highest IP address as that might be on a device with little CPU and memory available. The device you wish to be DR can be manually set to have a high DR priority number and problem devices can be set to a DR priority of zero, which means they can never be a DR or BDR. The BDR does exactly what it sounds like—it takes over DR duty if the DR fails.

OSPF also introduces the concept of **areas**. An area can be thought of as a mini-AS. Routers assigned to an OSPF area keep the topology map of that area only. A router that is only part of a single OSPF area is called an **internal** router. A Router that is part of multiple areas is called an **area border router (ABR)**. ABR's keep the topologies of all the areas in which they participate. An **AS boundary router (ASBR)** is a router in an AS which also participates in routing outside of the AS. These routers must run both OSPF and the external routing protocol, these days this external protocol is normally BGP. The OSPF area 0 is reserved for the OSPF backbone and should not be used for non-backbone devices. Backbone devices are typically the core and distribution layer devices at major WAN connected sites. Below is an example of an area 0, backbone area implementation (see **Figure 7**).

Figure 7.



SECTION 6

Internet Routing and BGP

Protocol such as OSPF and IS-IS work well in corporate or government networks as they exist today. These protocols will not scale to the level that is required to connect the vast number of networks now using the Internet. BPG is widely used to route traffic from AS to AS across the Internet. Very large networks with multiple autonomous systems also use BGP to route between their autonomous systems. The details of BGP routing are beyond the scope of this paper, but we will examine the functions BGP provides for Internet routing. BGP creates adjacencies much like OSPF does and then uses a complex process called **finite state machine (FSM)** to establish TCP sessions with adjacent BGP devices. Once routes are BGP established, it is critical that the routing database be stable throughout the entire BGP domain. Considering the sheer number of devices connected to the Internet, route flapping is likely to occur. If flapping by one or more AS border routers is left unchecked, the entire Internet could be slowed or halted while routing converged over and over again. BGP incorporates a special mechanism, **route flap damping**, to correct and eliminate route instability. The first time a route destination becomes unreachable then reachable in a short amount of time, BGP sets a damping timer. If this does not happen again before the timer expires, BGP releases the timer. If the destination flaps again while the timer is active, BGP places a hold on the route for a set period of time and resets the timer. Each time the route flaps after that, the router is held for an exponentially greater amount of time. After the flapping has subsided for a set amount of time, BGP allows the route back in.

Each individual AS has one or more registered Internet address that BGP uses to deliver all traffic destined for the AS. The AS routing protocol is responsible for delivery within the AS once BGP has delivered the packets to the ASBR. Because of this, BGP does not need to know the details of routing within the AS. This is analogous to the postal service delivering mail to different post offices using the zip code. The postal service does not have to worry about the details of delivery until the mail arrives at the office responsible for delivery within the marked zip code. Even with the use of AS roll ups, the global routing table used on the Internet had more than 300,000 entries in mid-2009, up from about 260,000 in mid-2008.

SolarWinds IP Address Manager (IPAM)

Are you still using spreadsheets to manage your IP space? SolarWinds IP Address Manager (IPAM) enables you and your team to ditch your spreadsheets and switch to easy-to-use, centralized IP address management software. Now it's easier than ever to manage and monitor Microsoft® DHCP and DNS, as well as Cisco® DHCP servers, all from a single, intuitive Web console.

With SolarWinds IPAM, you can:

- Centrally manage, monitor, alert, & report on entire IP infrastructure
- Maintain Microsoft® DHCP/DNS & Cisco® DHCP services from a single Web interface
- Optimize IP space utilization & avoid IP conflicts via automatic scans & preventative alerts
- Deliver role-based access control along with detailed event recording & activity logs
- Gain critical insight into IP address space through real-time views & historical tracking

Get more information on IP Address Manager here.

Evaluate IP Address Manager in a live demo environment here.

Download the fully functioning, 30-day evaluation here.

About SolarWinds

SolarWinds is rewriting the rules for how companies manage their networks. Guided by a global community of network engineers, SolarWinds develops simple and powerful software for managing networks, small or large. Our company culture is defined by passion for innovation and a philosophy that network management can be simplified for every environment.

SolarWinds products are used by more than one million network engineers to manage IT environments ranging from ten to tens of thousands of network devices. Comprised of fault and performance management products, configuration and compliance products, and tools for engineers, the SolarWinds product family is trusted by organizations around the globe to design, build, maintain, and troubleshoot complex network environments.

SolarWinds is headquartered in Austin, Texas, with sales and product development offices around the world. Join our online community of experts at thwack.com!

About the Author

Andy McBride is a Technical Specialist for SolarWinds focusing on making knowledge of networking and network management accessible to customers and prospects of all levels. The "Networking Fundamentals" series is specifically written for an audience with limited prior exposure to these technologies. Andy's technical background includes seven years at International Network Services (INS) as a Network Engineer and Managing Consultant, three years as a Novell Certified Instructor and five years as a Network Performance Products Manager with BT-Infonet. Prior to entering technology, Andy worked in aerospace on projects such as the SR-71, F-117, F-22, L-1011, F-18 and the space shuttle main engine. Andy has a degree in Chemistry but was wise enough to never work in that field. Andy invites you to follow him on Twitter, @McBrideA, and can be contacted on Thwack, McBrideA.