Networking Fundamentals

Part of the SolarWinds IT Management Educational Series

VOLUME 2

The Basics of Cisco IP SLA

This paper examines Cisco's IP SLA technology with an emphasis on network and device impacts as well as deployment strategies. As part of the Networking Fundamental series, it is intended for the user who is new to IP SLA.

solarwinds

Table of Contents

Document Revised: Oct 1, 2010

Introduction — The Case for Proxy Testing (IP SLA)

Let's start off by taking a look at a typical network management system (NMS) and how it operates. An NMS operates as a center-of-the-universe for network management. It keeps an inventory of devices and regularly tests those devices for availability and performance. The NMS also stores the results of these tests in a database and makes them available for viewing and report creation.

These are the three core functions of an NMS:

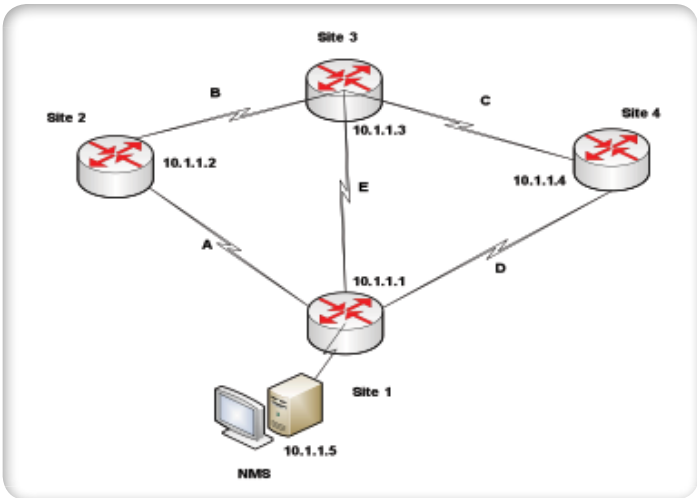
1. Data collection
2. Data storage
3. Data presentation

There are a couple of ways an NMS collects information about the network, including:

1. Performing tests directly; for example, the NMS can ping devices on the network.
2. Collecting information from a device by either:
   - Actively requesting the information from a device (polling) *Examples are SNMP polling and WMI polling*
   - Passively listening for data sent to the NMS (receiving) *Examples are SNMP trap reception and NetFlow data reception*

Consider the four-site network depicted below (**Figure 1**).

Figure 1. A four-site hybrid network



The NMS is installed in Site 1 and it uses ping to monitor the availability and delay of communications to the remote sites. The network pictured uses a simple routing protocol that will always choose the least number of hops to get from one site to another. This means if the NMS sends out a ping test to Site 4, the test will always take Link D if Link D is operational.

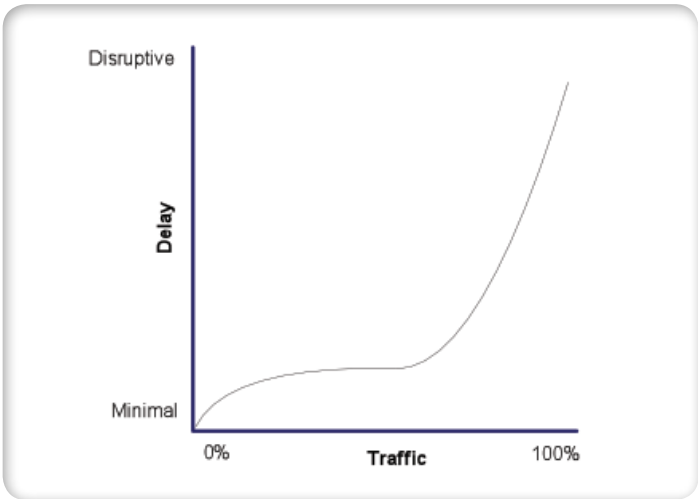
Here is how that test would occur:

1. NMS (10.1.1.5) sends out ping requests to 10.1.1.4.
2. The remote router receives the ping requests and sends a ping reply back to the NMS at 10.1.1.5.
3. The NMS receives the ping reply. This reply can contain a lot of information, but for our purposes, we'll just focus on two pieces of data:
   - Reachability — The successful ping reply from the remote router.
   - Performance — The Round Trip Time (RTT), or how long it took the ping response to be received by the NMS after the NMS sent the original ping request.

Assuming the ping test was successful, we now know that the remote router 10.1.1.4 is reachable from the NMS and we also know how many milliseconds it takes to send packets round trip between the NMS and the remote router. If we repeat this test to the other routers we have a good measure of the availability and performance of the network from the perspective of the NMS. This is why I referred to the NMS as a center-of-the-universe. At this point, all the NMS knows is what the network looks like from where it is installed.

Now, let's say Link C becomes 95% saturated with data for a prolonged period of time. This saturation will eventually result in slower response times between Sites 3 and 4, and this could be measured by ping tests between Sites 3 and 4. Now here's the problem: the NMS is at Site 1 and so can only test from Site 1. Assuming Link D is healthy, NMS ping tests would report delay to Site 4 as minimal. But, the users at Site 3 connecting to devices at Site 4 would be experiencing delay characterized in the below graph (**Figure 2**).

Figure 2. Delay as a function of traffic



The traffic would probably flow faster by routing through Link E rather than link D but our routing protocol decides the best route by least number of hops only. What we need is to measure the response between site 3 and site 4, and we need to find a way to get that information back to the NMS. This is what is known as **proxy testing**. IP SLA is a Cisco technology built into most Cisco devices for proxy testing. In the next section we'll look into the gears of IP SLA a bit.

How IP SLA Works

Cisco engineers recognized the need for proxy testing early on and released a feature called Response Time Reporter or RTR. This name sold the functionality short, as you'll see later, and the letters RTR were already well used in the industry as short hand for "router". D'oh!! Cisco later named the function Service Assurance Agent (SAA). This name was short-lived, and in 2004 Cisco renamed it again to IP SLA, or Internet Protocol Service Level Agreement, a much more descriptive name. Here we'll examine the mechanisms behind IP SLA.

First, let's get the nomenclature out of the way:

- In IP SLA, a **Source** is a device that creates and inserts IP SLA packets into the network.
- The **Target** is the ultimate destination of the packets.
- An **Operation** is the type of test being performed.

Looking at **Figure 1** again, the NMS can directly measure availability and ping response time to all three remote sites but has no direct method for testing the *quality* of the links throughout the network. Consider again the case where link C becomes heavily congested. The NMS can detect this high traffic level using SNMP polling of the interfaces at site 3 and site 4, but would be unable to determine if the congestion was causing performance impacts from site 3 to site 4. To do this, we need something at site 3 (or 4) to test the performance between the sites. We'll place an IP SLA operation on the Site 3 router targeting the site 4 router. Here's a config for the site 3 router (12.4(T) IOS), which will make the site 3 router the source, the site 4 router the target, and will implement the IP SLA ICMP Path Echo operation.

Router3 (config)# ip sla monitor 1

Router3 (config-ip-sla-monitor)# type echo protocol ipIcmpEcho 10.1.1.4

Router3 (config-ip-sla-monitor)# frequency 300

Router3 (config-ip-sla-monitor)# exit

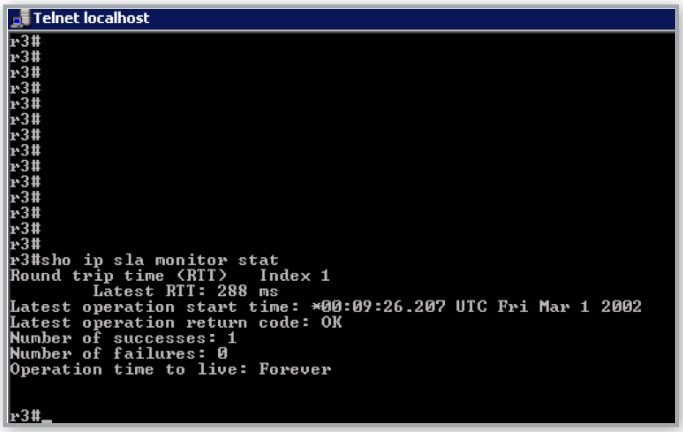
Router3 (config)# ip sla monitor schedule 10 life forever start-time now

Here's what we just did:

1. Entered "config ip sla monitor" config mode and assigned a number to the operation. (Each operation on a device needs a unique number, so I assigned this one as operation 1.)
2. Assigned the ICMP echo test type and target of 10.1.1.4.
3. Set the test frequency to 300 seconds.
4. Returned to base config mode.
5. Specified the operation schedule (never go away and start right away).

With this configuration in place, the router at site 3 will send ping (ICMP Echo requests) to site 4 every 300 seconds. We can examine the results of the operation using the *show ip sla monitor statistics* command directly off of the site 3 router in CLI. This will show us the last ping test result as seen below (**Figure 3**):

Figure 3. Screen output of IP SLA ping test



I have configured this operation between two Cisco routers but for an ICMP echo operation you can use anything that will respond to a ping as the target. Some operations are Cisco target specific and some are not. For a detailed explanation of each operation, see this Cisco resource guide. http://www.cisco.com/en/US/docs/ios/12_4/ip_sla/configuration/guide/hsla_c.html

Now we have the test running, but how do we get the information to the NMS? IP SLA capable devices store the results of the IP SLA operations in a Management Information Base (MIB) on the source device, in this case the RTTMON MIB. The NMS uses SNMP polling to retrieve the test results from the RTTMON MIB in the device and stores them in the NMS database. Now we can use the NMS to view reports about the delay from the NMS to each remote site, using the pings sent from the NMS and the delay between site 3 and 4 (and the other sites) based on the results of the IP SLA ICMP Path Echo operation.

Here is what we might see:

Site Availability and Delay (From NMS sent pings)

Site	Availability	Delay
Site 2	100%	21ms
Site 3	100%	19ms
Site 4	100%	22ms

IP SLA Results

Source	Destination	Operation	RTT	User Status
Site 1	Site 2	Ping	21ms	Happy
Site 1	Site 3	Ping	21ms	Happy
Site 1	Site 4	Ping	25ms	Happy
Site 3	Site 2	Ping	26ms	Happy
Site 3	Site 4	Ping	910ms	Red-faced

Here we have a much more accurate picture of the network than if we had relied on NMS pings only. The saturated link (C) has a delay of almost a full second! As you can see, this is well over the level required to cause red-faced users. Many applications will either become unreasonably slow or just plain fail. And real-time applications, like VoIP (Voice over IP), will be dead in the water.

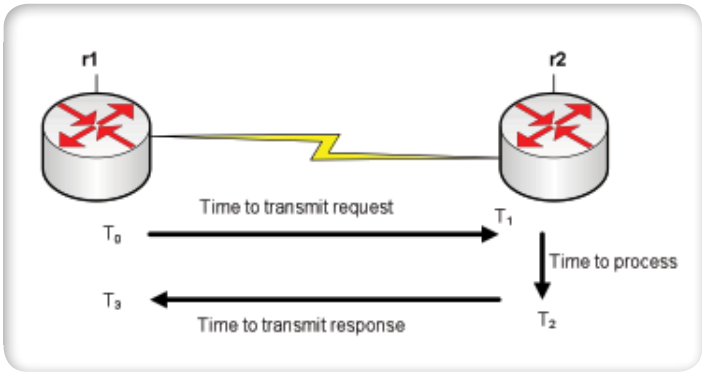
This is a rather simple example of IP SLA testing. There are many more operation types to look at, and some math to discuss along the way. But first, there is one more piece of IP SLA to discuss.

The Responder
(Wasn't that a Steven Segal movie?)

Ping tests are very simple and low-level. On many systems, the job of replying to pings is left to the network interface card or the forwarding plane of a router. The target's operating system doesn't have to be involved in processing each ping request. This means that the processing time at the target is minimal and the ping RTT (Round Trip Time) is a fairly good measure of the time-in-flight of the ping request and response. If we were to configure the UDP Echo operation, we are asking the target device to open a particular UDP port.

This action involves more sophisticated processing than ping and requires control plane and OS interaction. The processing time on the target device may be significant and if we are trying to measure only network delay it will skew the results. The IP SLA Responder was made to remove the target processing time from the final test value. **Figure 4** below depicts this interaction.

Figure 4. IP SLA Processing



The time to send the UDP echo, process the request at the target and return the result is $T_1 + T_2 + T_3$.

If we apply the IP SLA responder to r2, the UDP Echo IP SLA operation results at r1 for will be $T_1 + T_3$, subtracting out the processing time on r2. IP SLA responders can only be implemented on Cisco devices. Some operations will not use a responder, some require it and for others it is optional. For more information on the responder see http://www.cisco.com/en/US/technologies/tk648/tk362/tk920/technologies_white_paper09186a00802d5efe.html

What Do IP SLA Operations and Rabbits Have in Common?

It has to do with numbers. You know how rabbits can get out of hand... well, so can IP SLA operations. The trick is to deploy operations with some planning. Before we get into the planning strategies we'll take a look at how the numbers for IP SLA operations work. Consider the below network topologies (**Figures 5, 6, and 7**):

Figure 5. Hub and spoke network

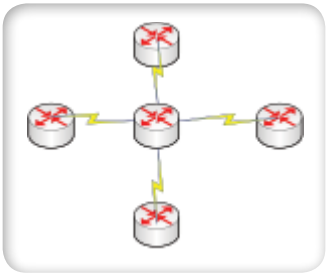


Figure 6. Hybrid network

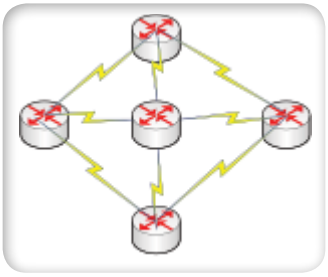
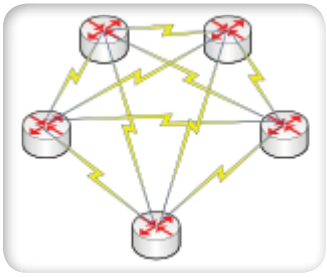


Figure 7. Full mesh network



Each of these topologies has a different number of connections between sites. The hub-and-spoke and full mesh networks are predictable in the following ways: in a hub and spoke, the main site (hub) is directly connected to each remote site (spoke).This is typical of a retail network where each point-of-sale store is a spoke, and the corporate headquarters is the hub. In the full mesh network, each site is directly connected to all other sites. This is typical of a high-performance network using VoIP and video between sites. Hybrid networks can be arranged in any fashion that best fits the needs of the network services, and so there is no predictable pattern for the number of connections required.

Here is how you can calculate the number of links in hub-and-spoke and full mesh networks where *L* = number of links and *N* = number of sites.

Hub and spoke: *L = N -1*

Full mesh: *L = N(N-1)/2*

So the number of links for a five site hub and spoke is: *L = (5-1)=4*

For full mesh we have: *L = 5(5-1)/2 = 5*4/2=10*

This is the number of links, but to test each link bi-directionally (site 1 to site 2 and site 2 to site 1) we need twice as many operations as links.

The equations for the number of operations (O) are

- Hub and spoke, *O = (N-1)2*
- Full mesh, *O = N(N-1)*

Therefore the total number of operations for the five site hub and spoke and five site full mesh are

- Hub and spoke, *O = (5-1)2 = 8*
- Full mesh, *O = 5(5-1) = 20*

If we add one more site to the hub and spoke we have 5 total links and 10 operations. If we add one more site to the full mesh we have 15 links and 30 operations. If we were to then add one more site to the mesh we have 21 links and 42 operations! Full mesh is the rabbit of network topologies. It is also the fastest growing of the WAN topologies due mostly to the ability to virtualize the connections at each site (so you don't need a physical WAN connection to each site), and the popularity of Multiprotocol Label Switching (MPLS).

Let's also consider that IP SLA offers several operation types.

Here is a list of some common operations you could apply:

ICMP Path echo (ping)

UDP echo

UDP Jitter

TCP connect

VoIP UDP Jitter

DHCP

HTTP

DNS

FTP

What if we deployed ping, UDP jitter, TCP connect, and UDP echo in each path of a seven site full mesh? We already know that for ping we would need 42 operations. For these four operation types to be tested throughout the mesh, we need to configure 42*4, or 168 operations. Now let's take a look at a typical mid-sized network with 30 sites. Here is the arithmetic:

*L=30*29/2 = 435*

*Total operations = 435*2*4 = 3480*

As you can see the numbers get rabbit-like very quickly. How about a large network with 180 sites?

*L=180*179/2 = 16,110*

*Total operations = 16,110*2*4 = 128,800!*

Obviously we need to step back from the mouse at some point and think about what we are measuring and why. By continuing to add operations, especially in a full mesh network, you could kill the network performance in the name of testing performance. This is where IP SLA planning comes into play.

IP SLA Deployment Strategies

Besides the possibility of burdening the network with test packets as we saw above, there are a few other things to consider about a large number of tests:

- Operations require resources on the target and the source. Large numbers of operations on a single device may impact the device performance by consuming CPU and memory.
- Data storage requirements: several thousand test results stored every five minutes can create a large database affecting other services on the NMS database.
- Viewing the results: chances are most of the historical results will never be examined.
- Polling engine: adding thousands of tests could add a significant burden to the SNMP poller.

So IP SLA can be dangerous when improperly implemented.

Here are strategies to avoid these issues:

1. **Keep local tests local. (Test locally, report globally!)**
   Not all test types are used to test WAN services. Take DHCP (Dynamic Host Configuration Protocol) for example. A large network may have several distributed DHCP servers. If each site has a local DHCP server, users at that site would receive IP addresses from the local server if it is available. For 40 sites you could accomplish DHCP testing by deploying an operation from the each site's local switch or router to the site's local DHCH server. 40 tests, 40 results to poll and store. You might also add tests for some secondary DHCP servers and have 50 or so total tests. If you added all DHCP testing to all sites to all servers you would have approximately 40^2 (1600) tests. Most of these tests are for DHCP requests to remote sites which will never actually be what the users request when obtaining an IP address.

2. **Test paths only for traffic they are expected to support.**
   Let's take the case of UDP jitter, a common IP SLA test. On an MPLS (Multiprotocol Label Switching), 40-site network, we'll implement the UDP jitter operation between the 5 sites that use UDP to deliver video conferencing. Since video conferencing is sensitive to network jitter and delay, it makes sense to implement jitter operations between these sites. Using the formula for a full mesh network (MPLS is always full mesh), we need to setup 20 operations. What if we just hit the "Full Mesh" button and deployed the tests between all sites. We would have 40*39/=1,560 tests and only 1.3% of the tests would be for valid video paths! Here we can see that a custom deployment of the operations is the wise thing to do.

3. **Consider decreasing the test frequency when possible.**
   The math on this is very straightforward. Decreasing the test frequency from 300 seconds to 360 seconds will lessen the test impact on the source device and network twenty percent. Increasing the frequency to 150 seconds will increase the load one hundred percent.

4. **Avoid overlapping tests.**
   We could deploy a DNS test to an internal DNS server, an HTTP test to an intranet page, a ping test to the HTTP server, and a TCP connect to the HTTP server from a local switch. While we have four individual tests testing four services, we actually have three arguably redundant tests. The HTTP operation does the following:
   - Resolves the URL to an IP address using the DNS server.
   - Makes a TCP port 80 request to the HTTP server.
   - Requests the HTTP and detects a successful page load.
   - Records the DNS resolve time, TCP open time and page load time.

   So with the HTTP test we can eliminate the other three tests and still get the same results.

Review

From top to bottom, here is an overview of the important concepts from the previous sections. If you have skipped the rest and started here, shame on you!

- IP SLA is a feature built into Cisco IOS
- IP SLA tests (Operations) are initiated from Cisco devices (Source) to measure the service quality of the path to another device (Target).
- Targets may be Cisco device or other types of devices. The target type may depend on the operation. For example, HTTP operations must be target at an HTTP server, Jitter operations must be targeted at a Cisco device, ping can target anything that will respond to ping.
- IP SLA responders (an IOS command) can be deployed to eliminate target processing time from an operation result.
- Applying operations to a network (especially a full-mesh WAN) must be done with caution! The number of operations created on an entire full-mesh is approximately the number of sites squared or N(N-1). Applying operation only where required is wise.
- Other measures that can be taken to minimize impact on the network include
  - Applying tests for local resources, like DHCP, only on local routers/switches
  - Decreasing test frequency
  - Avoid deploying test with overlapping results For example, HTTP and DNS tests.

I hope this has been helpful and that you will soon be taking advantage of IP SLA in your network!

Related SolarWinds Products

SolarWinds Solutions for IP SLA Monitoring

Now that you've got a solid understanding of Cisco IP SLA technology, we wanted to introduce a couple great solutions to help you begin using IP SLA on your network. Hey, you didn't think we'd miss out on the opportunity to show off our cool IP SLA tools, did you? Take a peek at these popular tools:

Orion IP SLA Manager

Orion IP SLA Manager evolved from our previous Orion VoIP Monitor to deliver a powerful solution for identifying site-specific and WAN-related network performance issues. This module identifies which devices on your network support IP SLA operations and automatically sets up operations, eliminating any guesswork. Finally, you can monitor key WAN applications by analyzing the performance of the underlying network protocols, including DNS lookups, FTP, HTTP, TCP connect, and UDP jitter. Of course, you can also continue to monitor VoIP call paths to ensure quality of service for your voice traffic.

Orion IP SLA Manager Highlights:

- Monitor WAN network performance using IP SLA technology that's already built into your existing Cisco routers
- Visualize site-to-site network performance on a clickable, drill-down map
- Discover and automatically setup Cisco IP SLA-capable network devices with specific IP SLA operations
- Quickly review WAN performance to determine the impact on key applications
- View at-a-glance WAN performance with the Top 10 IP SLA dashboard
- Monitor VoIP performance statistics, including MOS, jitter, network latency, and packet loss

Get more information about Orion IP SLA Manager here.

Evaluate Orion IP SLA Manager in a live demo environment here.

Download the fully functioning, 30-day evaluation here.

SolarWinds IP SLA Monitor Free Tool

- Analyze performance between a Cisco IP SLA-enabled device and other remote devices
- Monitor common IP SLA operations, including UDP echo, ICMP path echo (ping), TCP connect time, DNS resolution, and HTTP response
- Create and export a Universal Device Poller (UnDP) to monitor path-specific IP SLA performance in Orion Network Performance Monitor
- Verify and monitor quality of service (QoS)
- View IP SLA operation details, including frequency, source and target, operation type, and Type of Service (ToS) settings
- Prevent performance degradation by watching threshold-specific indicators to visually alert you when a performance problem occurs

Get more information about IP SLA Monitor here.

Download the free IP SLA Monitor tool 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.