



Practice tests



Video Training



Flash Cards



Study Planner

Official Cert Guide

Advance your IT career with hands-on learning

Cisco Certified DevNet Associate

DEVASC 200-901

Chris Jackson, CCIE® x2 (R&S & SEC) No. 6256

Jason Gooley, CCIE® x2 (R&S & SP) No. 38759

Adrian Iliesiu, CCIE® R&S No. 43909

Ashutosh Malegaonkar

ciscopress.com

Cisco Certified DevNet Associate DEVASC 200-901 Official Cert Guide

CHRIS JACKSON, CCIEX2 (RS, SEC) [CCIE NO. 6256]

JASON GOOLEY, CCIEX2 (RS, SP) [CCIE NO. 38759]

ADRIAN ILIESIU, CCIE RS [CCIE NO. 43909]

ASHUTOSH MALEGAONKAR

Cisco Press

Cisco Certified DevNet Associate DEVASC 200-901 Official Cert Guide

Chris Jackson, Jason Gooley, Adrian Iliesiu, Ashutosh Malegaonkar

Copyright© 2021 Cisco Systems, Inc.

Published by:
Cisco Press

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

ScoutAutomatedPrintCode

Library of Congress Control Number: 2020937218

ISBN-13: 978-01-3664296-1

ISBN-10: 01-3664296-9

Warning and Disclaimer

This book is designed to provide information about the Cisco DevNet Associate DEVASC 200-901 exam. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Editor-in-Chief: Mark Taub

Copy Editor: Catherine D. Wilson

Alliances Manager, Cisco Press: Arezou Gol

Editorial Assistant: Cindy Teeters

Director, ITP Project Management: Brett Bartow

Cover Designer: Chuti Prasertsith

Executive Editor: James Manly

Production Manager: Vaishnavi Venkatesan,
codeMantra

Managing Editor: Sandra Schroeder

Composition: codeMantra

Development Editor: Ellie Bru

Indexer: Ken Johnson

Technical Editors: Bryan Byrne, John McDonough

Proofreader: Donna Mulder

Project Editor: Lori Lyons




Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

 Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

About the Authors

Chris Jackson, CCIE No. 6256 (R&S and SEC), is a Distinguished Architect and CTO for Global Sales Training at Cisco. Chris is focused on digital transformation and showing customers how to leverage the tremendous business value Cisco technologies can provide. He is the author of *Network Security Auditing* (Cisco Press, 2010), *CCNA Cloud CLDADM 210-455 Official Cert Guide* (Cisco Press, 2016), and various online video courses for Cisco Press. He holds dual CCIEs in security and routing and switching, CISA, CISSP, ITIL v3, seven SANS certifications, and a bachelor's degree in business administration. Residing in Franklin, Tennessee, Chris enjoys tinkering with electronics, robotics, and anything else that can be programmed to do his bidding. In addition, he is a 3rd Degree Black Belt in Taekwondo, rabid *Star Wars* fan, and has a ridiculous collection of Lego. His wife Piper and three children Caleb, Sydney, and Savannah are the true joy of his life and proof that not everything has to plug into a wall outlet to be fun.

Jason Gooley, CCIE No. 38759 (R&S and SP), is a very enthusiastic and spontaneous person who has more than 20 years of experience in the industry. Currently, Jason works as a Technical Evangelist for the Worldwide Enterprise Networking Sales team at Cisco Systems. Jason is very passionate about helping others in the industry succeed. In addition to being a Cisco Press author, Jason is a distinguished speaker at Cisco Live, contributes to the development of the Cisco CCIE and DevNet exams, provides training for Learning@Cisco, is an active CCIE mentor, is a committee member for the Cisco Continuing Education Program (CE), and is a program committee member of the Chicago Network Operators Group (CHI-NOG), www.chinog.org. Jason also hosts a show called "MetalDevOps." Jason can be found at www.MetalDevOps.com, @MetalDevOps, and @Jason_Gooley on all social media platforms.

Adrian Iliesiu, CCIE No. 43909 (R&S), is a network engineer at heart with more than 15 years of professional IT experience. Currently, Adrian works as a Technical Leader with the Cisco DevNet Co-Creations team. During his career, Adrian has worked in several roles, including team leader and network, systems, and QA engineer across multiple industries and international organizations. When not working on innovative projects with customers and partners, Adrian advocates the advantages of network programmability and automation with a focus on enterprise and data center infrastructure. He is an established blog author, distinguished speaker at Cisco Live, and a recipient of the coveted Cisco Pioneer award. Adrian also appeared on Cisco TechWiseTV, Cisco Champion podcasts, and DevNet webinars. He holds a bachelor's degree in Electronics and Telecommunications from Technical University of Cluj-Napoca and a master's degree in Telecommunication Networks from Politehnica University of Bucharest.

Ashutosh Malegaonkar is a Cisco Distinguished Engineer, a senior technical contributor, and an industry thought leader. His experience spans across different technology domains: ISR Platforms, Voice, Video, Search, Video Analytics, and Cloud. Over two decades at Cisco, he has done two startups and has won several accolades, including the Pioneer awards. He has delivered several keynotes and talks at Cisco Connect and

Cisco Live. He has also been a Tech Field Day Speaker. With more than 25 years of professional experience, he currently leads the DevNet Co-Creations team whose mission is to co-create, innovate, and inspire alongside our strategic customers, partners, and developers. Ashutosh inspires those around him to innovate, and he is continually developing creative new ways to use software and Cisco APIs to solve real problems for our customers. He has a deep understanding of the breadth of Cisco products and technologies and where they can best be applied to serve our customers. Ashutosh has 16 approved patents and two publications.

Contents at a Glance

	Introduction	xxv
Chapter 1	Introduction to Cisco DevNet Associate Certification	2
Chapter 2	Software Development and Design	22
Chapter 3	Introduction to Python	58
Chapter 4	Python Functions, Classes, and Modules	86
Chapter 5	Working with Data in Python	106
Chapter 6	Application Programming Interfaces (APIs)	128
Chapter 7	RESTful API Requests and Responses	144
Chapter 8	Cisco Enterprise Networking Management Platforms and APIs	174
Chapter 9	Cisco Data Center and Compute Management Platforms and APIs	214
Chapter 10	Cisco Collaboration Platforms and APIs	254
Chapter 11	Cisco Security Platforms and APIs	300
Chapter 12	Model-Driven Programmability	340
Chapter 13	Deploying Applications	374
Chapter 14	Application Security	420
Chapter 15	Infrastructure Automation	448
Chapter 16	Network Fundamentals	482
Chapter 17	Networking Components	510
Chapter 18	IP Services	532
Chapter 19	Final Preparation	552
Appendix A	Answers to the “Do I Know This Already?” Quiz Questions	558
Appendix B	<i>DevNet Associate DEVASC 200-901 Official Cert Guide</i> Exam Updates	570
	Glossary	573
	Index	582

Online Elements

Appendix C	Study Planner
	Glossary



CHAPTER 18

IP Services

This chapter covers the following topics:

- **Common Networking Protocols:** This section introduces protocols that are commonly used in networks and that you should be familiar with.
- **Layer 2 Versus Layer 3 Network Diagrams:** This section covers different types of network diagrams.
- **Troubleshooting Application Connectivity Issues:** This section describes how to troubleshoot application connectivity issues.

This chapter covers IP services concepts. It starts with an introduction to common networking protocols that you are bound to use on a daily basis and then discusses them in detail:

- **Dynamic Host Configuration Protocol (DHCP)** is used to dynamically allocate IP configuration data to network clients so that the clients can get access to the network.
- **Domain Name System (DNS)** is a hierarchical naming system used mostly to resolve domain names to IP addresses.
- **Network Address Translation (NAT)**, while not a protocol per se, deals with translations between private IP networks and public, globally routable IP networks.
- **Simple Network Management Protocol (SNMP)** has been developed for remote network monitoring and configuration.
- **Network Time Protocol (NTP)** is used to synchronize date and time between devices on a network.

The chapter also provides a comparison between Layer 2 and Layer 3 network diagrams, as well as a troubleshooting scenario for application connectivity issues.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 18-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quiz Questions.”

6. What network devices should be included in a Layer 3 diagram?
 - a. Switches, routers, and firewalls
 - b. Routers, firewalls, and load balancers
 - c. Switches, firewalls, and load balancers
 - d. Bridges, switches, and hubs
7. What popular network utility is used to troubleshoot DNS issues?
 - a. traceroute
 - b. dnslookup
 - c. nslookup
 - d. ping

Foundation Topics

Common Networking Protocols

The following sections cover common networking protocols that network engineers and software developers alike should be familiar with. Knowledge of these protocols and technologies will give you better insight into how networks interact with applications in order to offer network clients an optimized and seamless experience.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP), as the name suggests, is a protocol used for dynamically configuring hosts with network connectivity information. In order for any host device connected to a network to be able to send and transmit data, it needs to have network parameters such as IP address, subnet, default gateway, and DNS servers configured. This configuration can be done either manually or automatically, using protocols such as DHCP. Manual configuration of network parameters for hosts on a network is time-consuming and prone to errors—and it is therefore not very often implemented in real-world networks anymore. DHCP is extensively used for automatically distributing network configuration parameters to all network endpoints, including end-user devices and network devices. For networks that have already migrated to IP version 6 (IPv6) or that are running dual-stack IP version 4 (IPv4) and IPv6 addressing, DHCPv6 and the IPv6 autoconfiguration option can be used for dynamic and automatic network parameter assignment. IPv6 autoconfiguration can be used to quickly and dynamically assign IPv6 addresses to network clients, and DHCPv6 is used to assign not just IPv6 addresses but also DNS servers and domain names.

DHCP has two components: a protocol for delivering network device configuration information from a DHCP server to a network host and a mechanism for allocating that configuration information to hosts. DHCP works using a client/server architecture, with a designated DHCP server that allocates IP addresses and network information and that delivers that information to DHCP clients that are the dynamically configured network endpoints.

Besides basic network connectivity parameters such as IP addresses, subnet masks, default gateways, IP addresses of DNS servers, and local domain names, DHCP also supports the concept of options. With DHCP options, a DHCP server can send additional configuration information to its clients. For example, Cisco IP Phones use option 150 provided by the DHCP server to obtain IP addresses of the TFTP servers that hold configuration files for the

IP Phones; Cisco wireless access points use DHCP option 43 to obtain the IP address of the Cisco wireless LAN controller that they need to connect to for management purposes.

DHCP for IPv4 is defined and described in RFC 2131: *Dynamic Host Configuration Protocol* and RFC 2132: *DHCP Options and BOOTP Vendor Extensions*. For IPv6, DHCP was initially described in RFC 3315: *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)* in 2003, but it was subsequently updated by several newer RFCs. RFC 3633: *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) Version 6* added a mechanism for prefix delegation, and RFC 3736: *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6* added stateless address autoconfiguration for IPv6.

Key Topic

Some of the benefits of using DHCP instead of manual configurations are

- **Centralized management of network parameters configuration:** DHCP servers usually manage the network configuration settings for several subnets and represent the central source of truth and the configuration point for all dynamic network parameters needed for network endpoints to be able to connect to the network. This makes it much easier to manage network address assignment compared to using several disparate systems or Excel files.
- **Reduced network endpoint configuration tasks and costs:** Dynamically allocating network connectivity information brings huge cost and time savings compared to manually performing the same tasks. This is especially true in medium to larger enterprise network environments and for Internet service providers (ISPs). Imagine ISPs needing to send out technicians to perform manual network changes to all of their customers when they start using their service every day. By using DHCP and dynamic network address configuration, the modems of clients can be configured within seconds, without any manual intervention.

DHCP is built on top of a connectionless service model using User Datagram Protocol (UDP). DHCP servers listen on UDP port 67 for requests from the clients and communicate with the DHCP clients on UDP port 68. There are several ways network configuration information is allocated by the DHCP server:

- **Automatic allocation:** With automatic allocation, the DHCP server assigns a permanent IP address to the client.
- **Dynamic allocation:** With dynamic allocation, the DHCP server assigns an IP address to the client for a limited period of time called the *lease time*.
- **Manual allocation:** With manual allocation, the network configuration of the client is done manually by the network administrator, and DHCP is used to relay that configuration information to the client.

As mentioned previously, DHCP defines a protocol and a process for how to assign network configuration information to devices connecting to the network. The process defines the methodology used to configure the DHCP server. Usually a DHCP server serves one or more client subnets. Once the client subnet is defined, a pool of addresses from that subnet is configured as available addresses for client allocation. Additional information such as subnet mask, the IP address of the default gateway, and DNS servers is the same for the whole subnet, so these configuration parameters apply to the whole subnet rather than to

each end host device. In some cases, the DHCP client and the server are located in different subnets. In such a case, a DHCP relay agent can be used to relay the DHCP packets between clients and servers. Any host on the network can act as a relay agent, but in most cases, the default router for the client subnet acts as a DHCP relay agent. Forwarding of DHCP messages between the clients and the servers by the relay agents is different from regular routing and forwarding. While regular forwarding is transparent for the endpoints involved in the exchange of data, with DHCP forwarding, DHCP relay agents receive inbound DHCP messages on the client interface and generate new DHCP messages on the interface connecting to the server. The DHCP relay agent effectively becomes a man-in-the-middle for the DHCP traffic between clients and servers. Figure 18-1 illustrates DHCP relay agent functionality.



Figure 18-1 *DHCP Relay Agent*

**Key
Topic**

DHCP operations fall into the following four phases (see Figure 18-2):

- Server discovery
- Lease offer
- Lease request
- Lease acknowledgment

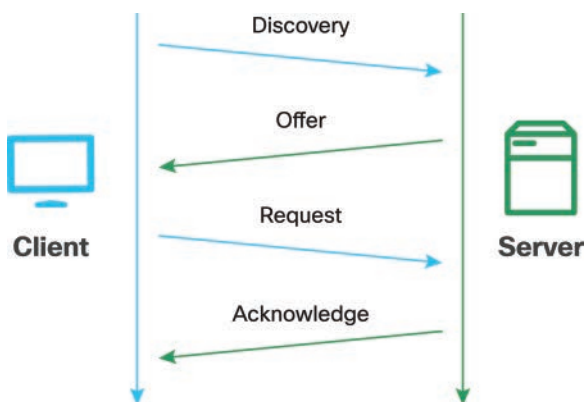


Figure 18-2 *DHCP State Machine*

Server Discovery

When a client first boots up and comes online on the network, it broadcasts a DHCP-
DISCOVER message with a destination address of the all-subnets broadcast address (255.255.255.255) with a source address of 0.0.0.0 since the client doesn't have any IP address at this stage. If there is a DHCP server configured for this subnet, it receives the all-subnets broadcast message and responds with a DHCPOFFER message containing the network

configuration parameters for the client. If there isn't a DHCP server directly configured on the same subnet as the clients but there is a DHCP relay agent, the agent forwards the request to the DHCP server and will also forward the server offer to the requesting client.

Lease Offer

A DHCP server that receives a DHCPDISCOVER message from a client responds on UDP port 68 with a DHCPOFFER message addressed to that client. The DHCPOFFER message contains initial network configuration information for the client. There are several fields in the DHCPOFFER message that are of interest for the client:

- **chaddr:** This field contains the MAC address of the client to help the client know that the received DHCPOFFER message is indeed intended for it.
- **yiaddr:** This field contains the IP address assigned to the client by the server.
- **options:** This field contains the associated subnet mask and default gateway. Other options that are typically included in the DHCPOFFER message are the IP address of the DNS servers and the IP address lease and renewal time.

Once the client receives a DHCPOFFER message, it starts a timer and waits for further offers from other DHCP servers that might serve the same client subnet.

Lease Request

After the client has received the DHCPOFFER from the server, it responds with a DHCPREQUEST message that indicates its intent to accept the network configuration information contained in the DHCPOFFER message. The client moves to the Request state in the DHCP state machine. Because there might be multiple DHCP servers serving the same client subnet, the client might receive multiple DHCPOFFER messages, one from each DHCP server that received the DHCPDISCOVER message. The client chooses one of the DHCPOFFER messages; in most implementations of the DHCP client, this is the first DHCPOFFER received. The client replies to the server with a DHCPREQUEST message. The DHCP server chosen is specified in the Server Identifier option field of the DHCPREQUEST message. The DHCPREQUEST message has as a destination address the all-subnets broadcast address once more, so all DHCP servers receive this message and can determine if their offer was accepted by the client. The source IP address of the DHCPREQUEST message is still 0.0.0.0 since the client has not yet received a confirmation from the DHCP server that it can use the offered IP address.

Lease Acknowledgment

The DHCP server receives the DHCPREQUEST message from the client and acknowledges it with a DHCPACK message that contains the IP address of the DHCP server and the IP address of the client. The DHCPACK message is also sent as a broadcast message. Once the client receives the DHCPACK message, it becomes bound to the IP address and can use it to communicate on the network. The DHCP server stores the IP address of the client and its lease time in the DHCP database.

Releasing

A DHCP client can relinquish its network configuration lease by sending a DHCPRELEASE message to the DHCP server. The lease is identified by using the client identifier, the chaddr field, and the network address in the DHCPRELEASE message.

Domain Name System (DNS)

Domain Name System (DNS) is a directory of networks that maps names of endpoints to IP addresses. DNS performs a critical role in all networks and especially on the Internet. It is much easier to remember the website `www.cisco.com` than it is to remember the IP address to which it resolves—`173.37.145.84` for IPv4 or `2600:1408:2000:1b3:0:0:b33` for IPv6. DNS is responsible for the process of resolving a hostname to an IP address. Each host endpoint and any device connecting to the network need to have an IP address configured in order to be able to communicate on the network. The IP address is like a street address, as every device on the Internet can be located based on its IP address. For example, when a user loads a web page, a translation must happen between the website name (`cisco.com`) and the machine-friendly IP address needed to locate that web page. This process is abstracted from the user because the user doesn't need to know what is happening in the background with the resolution of names into IP addresses.

Key Topic

There are several critical components in the DNS resolution process:

- The DNS recursive resolver is the server that receives DNS queries from client machines and is making additional requests in order to resolve the client query.
- Root name servers at the top of the DNS hierarchy are the servers that have lists of the top-level domain (TLD) name servers. They are the first step in resolving hostnames to IP addresses.
- TLD name servers host the last portion of a hostname. For example, the TLD server in the `cisco.com` example has a list for all the `.com` entries. There are TLD servers for all the other domains as well (`.net`, `.org`, and so on).
- The authoritative name server is the final step in the resolution process. It is the authoritative server for that specific domain. In the case of `cisco.com`, there are three authoritative servers: `ns1.cisco.com`, `ns2.cisco.com`, and `ns3.cisco.com`. Whenever a public domain is registered, it is mandatory to specify one or more authoritative name servers for that domain. These name servers are responsible for resolving that public domain to IP addresses.

Let's go through the steps of a DNS lookup from the perspective of a client that is trying to resolve a domain to an IP address (see Figure 18-3):

- Step 1.** The client query travels from the client machine to the configured DNS server on that machine. This DNS server is the DNS recursive resolver server.
- Step 2.** The DNS recursive resolver queries a DNS root name server.
- Step 3.** The root server responds to the recursive resolver with the TLD server for the requested last portion of the hostname. In the case of `cisco.com`, this would be the `.com` TLD server.
- Step 4.** The resolver queries the `.com` TLD server next.
- Step 5.** The TLD server responds with the IP address of the authoritative name server—in this example, the DNS server responsible for the `cisco.com` domain.
- Step 6.** The resolver sends the query to the authoritative name server.

- Step 7.** The authoritative name server responds with the IP address of the cisco.com web server.
- Step 8.** The DNS resolver responds to the client with the IP address obtained from the authoritative name server.
- Step 9.** The client can finally send the web page request to the IP address of the web server.
- Step 10.** The web server returns the web page to the client, and the browser renders it for the user.

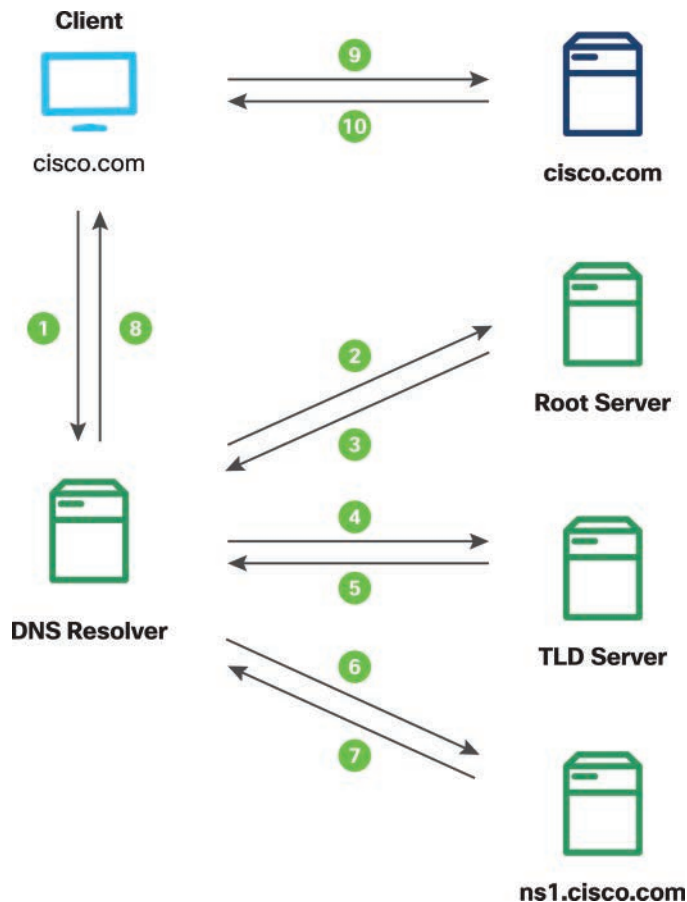


Figure 18-3 DNS Name Resolution Steps

A caching mechanism is available with DNS in order for the client queries to be resolved as quickly as possible. DNS caching means temporarily storing results obtained during previous requests on DNS servers that are close to the client. Caching DNS resolution data makes it possible to resolve client queries earlier in the DNS lookup chain, which improves resolution time and reduces bandwidth and CPU consumption.

**Key
Topic**

DNS uses User Datagram Protocol (UDP) on port 53 to serve resolution queries. Several different types of records are stored in the DNS database, including IP addresses (A records for IPv4 and AAAA records for IPv6 addresses), SMTP mail exchangers (MX records), IP addresses of name servers (NS records), and alias records (CNAME). Although it was not intended to be used as a general-purpose database, DNS has been extended to store many types of additional information.

The Internet Engineering Task Force (IETF) has published several Requests for Comments (RFCs) related to DNS over the years. Some of the most important ones are RFC 1034: *Domain Names—Concepts and Facilities*, RFC 1035: *Domain Names—Implementation and Specification*, and RFC 1123: *Requirements for Internet Hosts—Application and Support*.

Network Address Translation (NAT)

When Internet Protocol (IP) was created, very few people, if any, were expecting it to support a global network of billions of interconnected devices. As discussed in earlier chapters, IPv4 addresses are 32 bits long, which means they can uniquely address a bit more than 4 billion endpoints. This number was fine and out of reach for a long time, but as the number of endpoints connecting to the Internet grew exponentially, it was clear that 4 billion addresses would not be enough to uniquely identify all the connected devices. At that point, work started for a new version of IP, IPv6, which defines 128-bit addresses and is able to uniquely identify trillions of endpoints. At the same time, it was clear that an overnight switchover from one IP version to another would be an impossible feat on the Internet, so several temporary solutions were proposed to ease the transition and extend the life of the IPv4-based Internet.

**Key
Topic**

Network Address Translation (NAT) is one of the solutions to preserve the dwindling number of public IPv4 addresses. NAT reuses private IPv4 address blocks in internal networks and translates those addresses into public and unique IPv4 addresses at the borders of the internal networks. RFC 1918: *Address Allocation for Private Internets* declared a set of subnets private and unroutable on the global Internet. The subnets 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 are extensively used in all private networks in the world, from enterprise networks to small office/home office networks. All other IPv4 addresses are public and routable on the Internet, meaning they uniquely identify endpoints on the network.

NAT is mostly used to translate between private RFC 1918 subnets and public IPv4 subnets. This translation happens at the exit points from the private networks, which in most cases are firewalls or border routers. NAT can also be used to translate between private and private networks. In the case of mergers and acquisitions, it is possible that the enterprise that was acquired uses the same private IPv4 subnets as the acquiring company. In order to be able to exchange traffic between these networks that are addressed with the same IP addresses, NAT can be used to perform address translation. Basically, whenever an IP address needs to be translated into another address, NAT can be used.

**Key
Topic**

NAT is an IETF standard described in RFC 1631: *The IP Network Address Translator (NAT)*. A large number of Cisco and third-party routers and firewalls support NAT. The devices that perform IP address translations are usually situated and route data traffic between the internal network and the outside world or the public Internet. During the NAT configuration phase, an internal subnet or a set of subnets is defined as being internal, or

“inside”; these are usually the private IP subnets used internally in the enterprise. As a second step in the configuration process, single public IP address or an external, or “outside,” pool of IP addresses are defined. With this information, the border device can perform IP address translation between the internal and external worlds. Several types of NAT are available:

- **Static NAT (static NAT):** Static NAT defines a one-to-one mapping between the internal IP address and its public IP address correspondent.
- **Dynamic NAT (dynamic NAT):** With dynamic NAT, the internal subnets that are permitted to have outside access are mapped to a pool of public IP addresses. The pool of public IP addresses is generally smaller than the sum of all the internal subnets. This is usually the case in enterprise networks, where public IPv4 addresses are scarce and expensive, and a one-to-one mapping of internal to external subnets is not feasible. Reusing the pool of public IP addresses is possible as not all internal clients will access the outside world at the same time.
- **Port Address Translation (PAT or overloading):** PAT takes the dynamic NAT concept to the extreme and translates all the internal clients to one public IP address, using TCP and UDP ports to distinguish the data traffic generated by different clients. This concept is explained in more detail later in this chapter.

The type of NAT used in a particular situation depends on the number of public IP addresses defined and how the translation process is implemented.

In order to illustrate how NAT works, let’s assume that a client connected to an enterprise network uses private IPv4 addressing. As the client generates data traffic and tries to connect to the Internet, the traffic makes its way to the enterprise border device. The border device looks up the destination of the traffic and the NAT configuration. If the client IP address is part of the internal subnets that have to be translated, it creates an entry in its NAT table with the source and destination IP addresses, it changes the source IP address of the packet from the internal private IP address to the public IP address, and it forwards the packet toward its destination. As the data traffic is received at the destination, the destination node is unaware that the traffic received went through the NAT process. The IP addresses in the response traffic are swapped, and as the data traffic is being received by the border device, a lookup in the NAT table is done for the entry that was created as the traffic was exiting the network. The entry is matched, and the translation is done again—but this time in the reverse order, from the public IP address back to the private IP address of the client—and the traffic is routed back to the original source.

Key Topic

With PAT, the same type of table that keeps track of private to public translations and vice versa is created on the border device—but in this case TCP and UDP ports are also taken into account. For example, if the client generates web traffic and is trying to reach a web server on the Internet, the randomly generated TCP source port and the destination TCP port 443 for HTTPS are also included in the network translation table. In this way, a large number of clients—up to theoretically 65,535 for TCP and 65,535 for UDP traffic—can be translated to one public IP address. Figure 18-4 illustrates PAT, which is also called *overloading* because many internal private IP addresses are translated to only one public IP address.

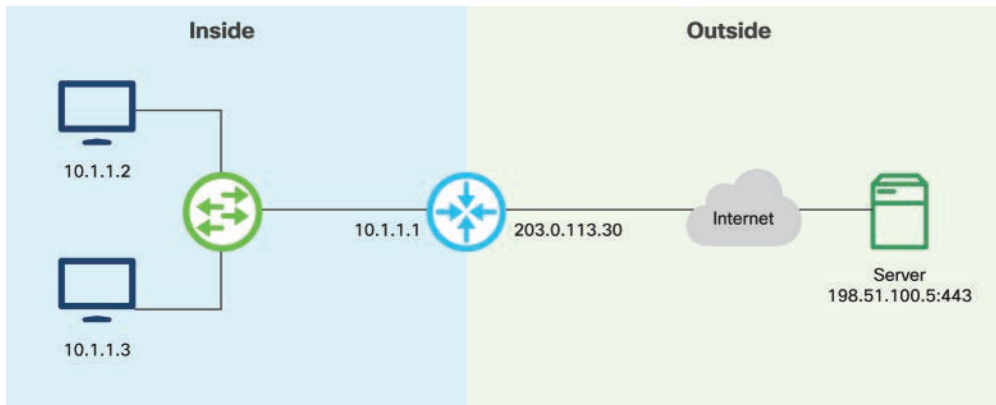


Figure 18-4 *Port Address Translation*

Some of the benefits of NAT are as follows:

- **Reduced costs for renting public IPv4 addresses:** With dynamic NAT and especially with PAT, a large number of private internal endpoints can be hidden behind a much smaller number of public IP addresses. Since public IPv4 addresses have become a rare commodity, there is a price for each public IPv4 address used. Large cost savings are possible by using a smaller number of public addresses.
- **Conserving public IPv4 address space:** The Internet would not have witnessed the exponential growth of the past few decades without NAT. Extensively reusing RFC 1918 private IP addresses for internal networks helped slow the depletion of IPv4 address space.
- **Additional security due to hiding the addressing for internal networks:** Having a whole network hidden behind one or a pool of public IPv4 addresses thwarts network reconnaissance attacks and increases the defense capabilities of the network.
- **Flexibility and cost savings when changing ISP connectivity:** In cases in which external network connectivity needs to be changed and migrations to new ISPs are required, configuration changes need to be performed only on the network border devices, but the rest of the internal network does not need to be re-addressed. This results in massive cost savings, especially in large enterprise networks.

Although the advantages of NAT, far outweigh the disadvantages in most cases, there are some disadvantages, including the following:

- **Loss of end-to-end functionality:** Some applications, especially for real-time voice and video signaling, are sensitive to changes in IP header addressing. While establishing these types of real-time voice and video sessions, headers exchanged at the application layer contain information pertaining to the private non-globally routable IP addresses of the endpoints. When using NAT in these cases, the IP addresses in the Layer 3 headers differ from the IP addresses contained in the application layer headers. This results in an inability to establish end-to-end voice and video calls.
- **Loss of end-to-end traceability and visibility:** Troubleshooting end-to-end connectivity issues is especially challenging in networks that use NAT.

- **Degradation of network performance:** NAT operations on border devices are usually not resource intensive, and several mechanisms have been implemented to make this impact even lower. Still, a border device needs to take an additional step before forwarding the data traffic toward its destination, and that means additional delay and consumption of memory and CPU resources.

NAT is extensively used in networks that use IPv4 addressing. One of the requirements for IPv6 was to restore end-to-end connectivity between all endpoints on the network, so NAT is not popular in IPv6 networks.

Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is an application layer protocol used for monitoring and configuring devices. It was originally developed in the 1980s, and the IETF has published several RFCs covering SNMP since then. As networks were becoming larger and more complicated in those days, a need to be able to remotely monitor and manage devices arose. Following are several versions of SNMP that have been released through the years:

- SNMP version 1 (SNMPv1)
- SNMP version 2 (SNMPv2)
- SNMP version 2c (SNMPv2c)
- SNMP version 3 (SNMPv3)

While versions 1 and 2 of SNMP are rarely used anymore, versions 2c and 3 are extensively used in production environments. SNMPv2 adds 64-bit counter support and includes additional protocol operations. With SNMPv3, the focus is on security, so additional features like authentication, encryption, and message integrity were added to the protocol specification.



The following are some of the advantages of SNMP:

- It provides a single framework for monitoring many different kinds of devices.
- It is based on open standards documented in IETF RFCs.
- It is easily extensible.

There are also disadvantages with SNMP, including the following:

- The lack of writable MIBs (Management Information Bases) leads to poor configuration capabilities. SNMP is rarely used for configuration purposes.
- The lack of atomic transactions makes rollbacks to previous states difficult.
- SNMP is slow for monitoring purposes when large amounts of operational data need to be retrieved.
- It is CPU and memory resource intensive when large amounts of performance metrics are retrieved.

Even though SNMP was originally designed to support both monitoring and configuration capabilities, historically only the monitoring capabilities were used. The SNMP specification defines the following three components:

- Managed devices
- SNMP agent
- SNMP manager

Managed devices are the devices that are being monitored and managed through SNMP. They implement an SNMP interface through which the SNMP manager monitors and controls the device. The SNMP agent is the software component that runs on the managed device and translates between the local management information on the device and the SNMP version of that information. The SNMP manager, also called the Network Management Station (NMS), is the application that monitors and controls the managed devices through the SNMP agent. The SNMP manager offers a monitoring and management interface to network and system administrators. The SNMP components and their interactions are illustrated in Figure 18-5.

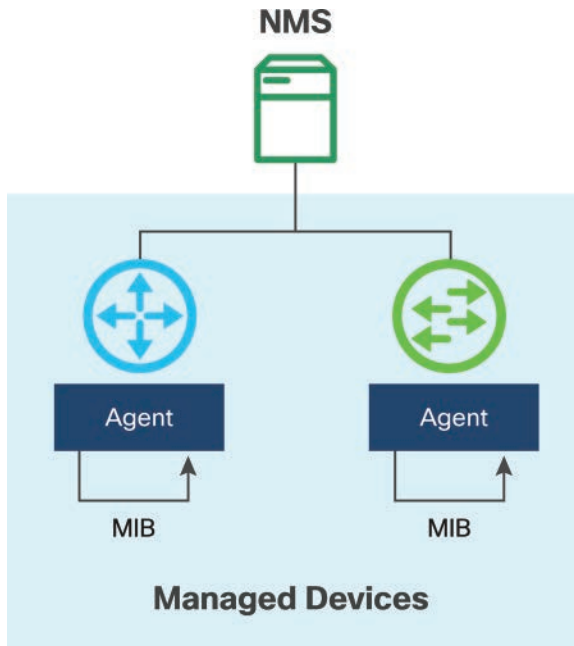


Figure 18-5 *SNMP Components*

**Key
Topic**

The SNMP agent listens on UDP port 161 for requests from the SNMP manager. SNMP also supports notifications, which are SNMP messages that are generated on the managed device when significant events take place. Through notifications, the SNMP agent notifies the SNMP manager about these critical events. The NMS listens for these notifications on UDP port 162. SNMP data structures that facilitate the exchange of information between the SNMP agent and the NMS are organized as a list of data objects called a Management Information Base (MIB). A MIB can be thought of as a map of all components of a device that are

being managed by SNMP. In order to be able to monitor devices, the NMS must compile the MIB file for each device type in the network. Having the appropriate MIB, the SNMP agent and manager can exchange a wide range of information.

A MIB is organized as a tree-like structure with unique variables represented as leaves. Each variable in the tree is uniquely identified by an Object Identifier (OID). Operational data such as CPU temperature, fan speed, and outbound packets on an interface are all represented through OID values. In order for the NMS to obtain any device's operational metric, it needs to send an SNMP GET packet that includes OID values for each metric of interest. The SNMP agent receives the packet and looks up the OIDs in the MIB, and if the device implements the requested information, it returns the information to the NMS.

Key Topic

Several types of SNMP messages are used to communicate between the SNMP manager and the agent:

- **GetRequest:** This is the type of message used by the NMS to request the value of a variable or list of variables. The agent returns the request information in a Response message.
- **SetRequest:** The SNMP manager uses this message to request a change to a variable or a list of variables. The agent returns a Response message containing the new values for the variables.
- **GetNextRequest:** The SNMP manager uses this message to discover available variables and their values. This is a common operation used to “walk” the entire MIB of an agent. The agent returns a Response message that contains the requested information.
- **GetBulkRequest:** This optimization of the GetNextRequest message, which was introduced with SNMPv2, contains multiple iterations of the GetNextRequest call.
- **Response:** The SNMP agent generates this message, which contains the information the SNMP manager requested.
- **Trap:** The SNMP agent generates this notification message to signal to the SNMP manager when critical events take place on the managed device.
- **InformRequest:** The SNMP agent sends this acknowledged notification to the SNMP manager. Keep in mind that SNMP runs over UDP, which is a connectionless protocol, and packets might get lost during transmission. A rudimentary acknowledgment mechanism is implemented through InformRequest messages.

SNMP community names are used to establish trust between managers and agents. When community names are configured, the SNMP requests from the manager are considered valid if the community name matches the one configured on the managed device. If the names match, all agent-based MIB variables are made accessible to the manager. If they do not match, SNMP drops the request.

Network Time Protocol (NTP)

Accurately keeping track of time is critical in today's IT infrastructure. As IT infrastructure becomes more and more ingrained in business processes and success, every second of downtime when the infrastructure is not available translates into loss of revenue for the business. In extreme cases, this can lead to loss of customers and bankruptcy. Service-level

agreements (SLAs) are contracts between providers and consumers of infrastructure. As an example, stringent SLA contracts require 99.999% uptime, which in the case of Internet service providers translates into no more than 5 minutes of downtime per year. It is of critical importance to make sure that there is a consistent, uniform, and correct view of time on all the devices in a network. Time is fundamental when measuring SLAs and enforcing contracts. Inaccurate time can lead to service disruptions. As a simple example, with web traffic, HTTPS TLS connections will not even be established if the time on the web server or the client is not accurate.

The system clock on each device is the heart of the time service. The system clock runs from the second the operating system starts; it keeps track of the date and time. The system clock can be set to update from different time sources and can be used to distribute time to other devices. Network Time Protocol (NTP) enables a device to update its clock from a trusted network time source and serve time to other devices, enabling groups of devices to be time synchronized. Most devices contain battery-powered clocks that keep track of date and time across restarts and power outages.

**Key
Topic**

The main role of NTP is to synchronize the time on a network of devices. It was developed by IETF, and the latest version of the protocol, version 4, is defined in RFC 5905: *Network Time Protocol Version 4: Protocol and Algorithms Specification*. NTP uses UDP at the transport layer, and port 123 is reserved for it. NTP works based on a client/server architecture, with NTP servers providing the time to clients. Authoritative time sources are servers that have attached radio clocks or atomic clocks, making them extremely accurate. NTP has the role of distributing time to all the devices connected to the network. Multiple NTP servers can coexist at the same time on the same subnet, and clients can use all of them for time synchronization. NTP clients poll the time servers at intervals managed dynamically by conditions on the network such as latency and jitter. One NTP transaction per minute is sufficient to synchronize time between two machines.

The concept of strata is used in NTP to describe how many hops or devices away a client is from an authoritative time source. NTP servers that are most authoritative and are directly connected to very accurate time sources are in stratum 1. A stratum 2 time server receives time from a stratum 1 server, and so on. When a client receives time from different NTP servers, a lower-stratum server is chosen as the most trusted source unless there is a big time difference between the lower-stratum server and all the other servers.

**Key
Topic**

There are two ways communication between NTP clients and servers takes place: IT can be statically configured or can occur through broadcast messages. You can manually configure NTP clients to establish a connection and to associate and solicit time updates from NTP servers by simply statically configuring the hostname or the IP addresses of the servers. In local-area networks within the same subnet, NTP can be configured to use broadcast messages instead. Configuration in this situation is simpler, as each device can either be configured to send or receive broadcast messages. With broadcast NTP messages, there is a slight loss of accuracy since the flow of information is only one way.

Since time accuracy is critical in today's infrastructure, it is recommended to implement all security features that come with NTP. Two NTP security features are most commonly used:

- An encrypted authentication mechanism between clients and servers should always be enabled.
- NTP associations should be limited to only trusted servers through access control lists.

In most situations, it is recommended to have at least three higher-stratum NTP servers configured for each network. A large number of public NTP servers can be used for these purposes.

Layer 2 Versus Layer 3 Network Diagrams

Network diagrams are a critical component in the documentation process for any network. Just as software developers document their code for easier maintenance and sharing, network engineers document networks by building network diagrams. It is very important to have accurate network diagrams, especially when troubleshooting network issues. The network has become a business differentiator for all companies, and each second of downtime costs money. Having up-to-date network diagrams makes it much easier and faster to troubleshoot network issues and decrease the MTTR (mean time to resolution) when problems arise.

Several different types of network diagrams can be created, depending on what needs to be emphasized and documented. Layer 1 network diagrams can be used to show physical connections, including how network devices are connected and what type of cables (twisted pair, fiber optics, and so on) are being used. These diagrams can also show patch panel port connectivity and availability and everything else that is in the confines of physical connectivity.

Key Topic

Layer 2 network diagrams contain information related to all Layer 2 devices, protocols, and connectivity in a network. Such a diagram shows all the switches in the network and how they are interconnected, including how the ports on one switch connect to the ports on another switch, which VLANs are configured on each switch, which ports are configured as trunks and what VLANs are allowed on them, which ports are configured as port channels, and any other Layer 2 information. Depending on the size of the network, a Layer 2 diagram can be split into multiple documents.

Key Topic

A Layer 3 network diagram captures all the network information available at Layer 3. Such a diagram should show all devices that operate at Layer 3—routers, firewalls, load balancers, and so on—and how they are interconnected. Information about IP addressing and subnets, routing protocols, first-hop redundancy protocols (such as HSRP, VRRP, and GLBP), Layer 3 port channels, and Internet connectivity should also be included at a minimum. More information can also be included in these diagrams but care should be taken to avoid including too much information or dynamic information that is bound to change often.

Several software tools are available for creating network diagrams. You can create network diagrams manually by using tools such as Microsoft Visio and draw.io, or you can have them created automatically with controller-based solutions such as Cisco DNA Center. With Cisco DNA Center, the controller automatically discovers all the devices in the network and how they are interconnected, and it uses this information to build network diagrams, topologies, and databases.

It is a good idea to store network diagrams in version control systems such as Git. Networks are dynamic and evolve over time, and it is important to capture this evolution in the diagrams. Git offers a proven solution to storing and keeping track of changes in documents, so it is ideal for storing network diagram versions. With infrastructure as code and solutions such as Cisco VIRT, network diagrams and topologies are stored as YAML files and can be used as part of CI/CD infrastructure configuration automation pipelines.

Troubleshooting Application Connectivity Issues

Applications are becoming more and more complicated, with many moving components that need to communicate with each other over the network. With the advent of the microservices architecture for applications and the ubiquity of APIs, the network is critical to the functionality of applications. It is important to know how to troubleshoot connectivity issues and how they might affect the performance of applications.

Network connectivity may not function as expected for a variety of reasons. Whenever an application or a website or any target destination that is being accessed over a network is not behaving as expected, the network takes the blame. We will see next that while the network might be the culprit in some instances, there are many other reasons applications stop responding as expected.

Key Topic

Network troubleshooting usually follows the OSI layers, discussed in Chapter 16, “Network Fundamentals.” It can occur from top to bottom, beginning at the application layer and going down the layers all the way to the physical layer, or it can occur from bottom to top. This section looks at a typical bottom-to-top troubleshooting session that starts from the physical layer and goes up the stack toward the application layer. The troubleshooting steps discussed here can also be followed in the reverse order, if desired.

First and foremost, it is important to know how an endpoint device connects to the network at the physical layer: with a wired or wireless connection. If the connection to the network is wired through an Ethernet cable, the network interface card should come online, and electrical signals should be exchanged with the switch port to which the NIC is connected. Depending on the operating system of the client connecting to the network, the status of the connection will show as solid green or will display as “enabled” or “connected” in the network settings. If the NIC status shows as connected, the physical layer is working as expected, and the troubleshooting process can proceed to the next layer. If the NIC doesn’t show as connected or enabled, there could be several causes, including the following:

- Misconfigured or disabled switch port
- Defective network cable
- Defective wall network port
- Incorrect cabling in the patch panel
- Defective network interface card

Troubleshooting at the physical layer revolves around making sure there is an uninterrupted physical connection between the client and the switch port to which it connects.

If the connection to the network is wireless, it is important to ensure that the wireless network interface card is turned on and that it can send and receive wireless signals to and from the nearest wireless access point. Being within the service range of a wireless access point is also important, and usually the closer the client is to the access point, the better network performance it should experience.

Key Topic

Troubleshooting at the data link layer, or Layer 2, means making sure the network client and the switch are able to learn MAC addresses from each other. On most operating systems, the client can check the MAC address table with the `arp` command, and on most Cisco switches,

the client can check the MAC address table with the **show mac address-table** CLI command. If the ARP table on both the client and the switch get dynamically populated with MAC address information, it means the data link layer is functioning as expected. Some of the issues that cause the data link layer not to function as expected are as follows:

- Misconfigured switch port
- Layer 2 access control lists
- Misconfigured Spanning Tree Protocol
- Missing or misconfigured VLANs

Key Topic

At Layer 3 (the Internet layer), IP connectivity and reachability have to work. The network client should be configured with the correct IP address, subnet mask, and default gateway. This is usually done using DHCP servers, and in rare cases it may be manually configured. Built-in operating system tools such as **ifconfig** and **ping** can be used at this layer to help with troubleshooting. **ifconfig** (or **ipconfig** on Microsoft Windows) is a network utility that retrieves and can modify the configuration and status of all network interfaces present on the client endpoint. **ping** is another popular network tool that is extensively used to verify endpoint IP reachability. If the destination of the client data traffic is in a different subnet than the client is connected to, that means that the traffic has to be routed through the default gateway of the client subnet. Layer 3 connectivity between data traffic source and destination can be checked using the **ping** command. If IP connectivity is verified end to end between source and destination, troubleshooting can proceed to the next step. If not, you need to look for the problems that can cause connectivity issues at Layer 3, including these:

- Misconfigured IP information (IP address, subnet mask, default gateway) on the client device
- Layer 3 access control lists that blocks data traffic
- Routing protocol issues causing black-holing or incorrect routing of traffic

Key Topic

Troubleshooting at the transport layer means making sure that the network clients can access the TCP or UDP ports on which the destination applications are running. For example, in the case of web traffic, it is important to verify that the client can connect to TCP ports 80 (HTTP) and/or 443 (HTTPS) on the web server. In some cases, web servers are configured to listen on esoteric ports such as 8080, so it is important to know the correct port on which the destination application is running. Networking tools such as **curl** and custom **telnet** commands specifying the application port can be used to ensure that transport layer connectivity can be established end to end between the source and destination. If a transport layer connection cannot be established, you need to look for issues such as these:

- Firewall access control lists blocking data traffic based on TCP and UDP ports
- Misconfigured applications and listening ports
- Misconfigured load balancers
- Presence of proxy servers that are intercepting the traffic and denying connectivity
- Misconfigured PAT

Other common problems that affect application connectivity are DNS related. As discussed earlier in this chapter, DNS plays a critical role in resolving domain names to IP addresses. If DNS is malfunctioning for some reason, end-to-end connectivity is impacted. Network tools such as **nslookup** can be used to troubleshoot DNS functionality. The following problems commonly cause DNS issues:

- Misconfigured DNS resolver on the network client
- Wrong hostname specified
- Invalid DNS server configuration
- Missing or incorrect DNS entry

**Key
Topic**

Even if end-to-end connectivity at the transport layer is verified, there can still be issues on the network that cause connections to applications to fail. These issues are usually rather difficult to discover and troubleshoot and are related to data traffic load and network delay. The difficulty with these issues comes from the fact that they are difficult to reproduce and can be temporary in nature, caused by short spikes in network traffic. Networking tools such as **iperf** can be used to dynamically generate traffic and perform load stress on the network to ensure that large amounts of data can be transported between the source and destination. Implementing quality of service (QoS) throughout the network can help with these problems. With QoS, traffic is categorized in different buckets, and each bucket gets separate network treatment. For example, you can classify traffic such as voice and video as real-time traffic by changing QoS header fields in the Layer 2 data frames and Layer 3 packets so that when switches and routers process this type of traffic, they give it a higher priority and guaranteed bandwidth, if necessary.

At the application layer, network tools such as **tcpdump** can be used to capture actual data traffic received on any of the network interfaces of either the source device or the destination device. Comparing between sent and received data can help in troubleshooting connectivity issues and determining the root cause of a problem. Slow or no responses at the application layer could indicate an overloaded backend database, misconfigured load balancer, or faulty code introduced through new application features.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 19, “Final Preparation,” and the exam simulation questions on the companion website.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 18-2 lists these key topics and the page number on which each is found.

**Key
Topic**
Table 18-2 Key Topics

Key Topic Element	Description	Page Number
List	Benefits of DHCP	535
List	Four phases of DHCP operations	536
List	DNS resolution process components	538
Paragraph	User Datagram Protocol (UDP)	540
Paragraph	Network Address Translation (NAT)	540
Paragraph	IETF NAT, described in RFC 1631	540
Paragraph	Port Address Translation (PAT)	541
List	Advantages of SNMP	543
Paragraph	SNMP agent listening for requests from the SNMP manager	544
List	SNMP message types	545
Paragraph	Main role of NTP	546
Paragraph	Two forms of communication between NTP clients and servers	546
Paragraph	Layer 2 network diagrams	547
Paragraph	Layer 3 network diagrams	547
Paragraph	Network troubleshooting	548
Paragraph	Troubleshooting at the data link layer	548
Paragraph	Troubleshooting at the Internet layer	549
Paragraph	Troubleshooting at the transport layer	549
Paragraph	Network issues with application connection failures	550

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), top-level domain (TLD) name server, Network Address Translation (NAT), Port Address Translation (PAT), Simple Network Management Protocol (SNMP), Network Management Station (NMS), Management Information Base (MIB), Object Identifier (OID), Network Time Protocol (NTP)