**Linux Server vs. Windows Server: The Main Differences**

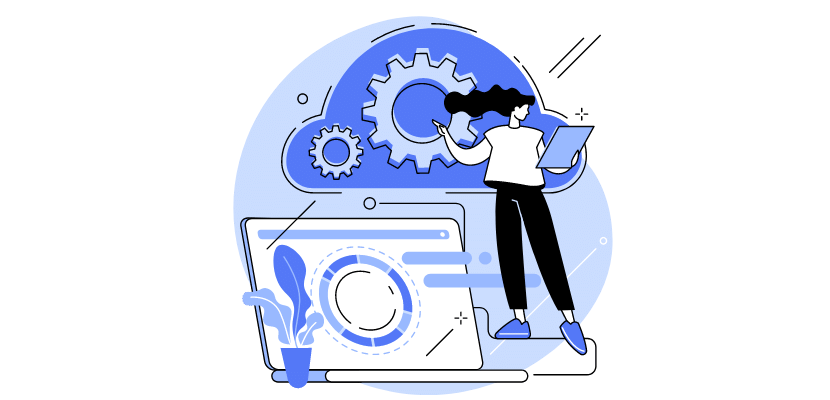
|  | **LINUX SERVER** | **WINDOWS SERVER** |
| --- | --- | --- |
| **ARCHITECTURE** | centered around the Linux kernel | based on the Windows NT architecture |
| **COST** | free, [open-source software](https://phoenixnap.com/glossary/what-is-open-source) | owned by Microsoft, includes a licensing fee per user |
| **SECURITY** | highly secure against malware and cyber threats | more prone to hacking attempts and cyber threats |
| **SUPPORT** | large community supports that can answer commonly asked questions | community and long-term customer support, along with great documentation |
| **MODE OF OPERATION** | command line | [graphical user interface](https://phoenixnap.com/glossary/what-is-gui) |
| **USER EXPERIENCE** | requires an relatively experienced Linux administrator | more beginner-friendly |
| **DATABASE SUPPORT** | MySQL, PostgreSQL | Microsoft SQL, Microsoft Access |
| **SCRIPT SUPPORT** | Python, PHP, [Perl](https://phoenixnap.com/glossary/what-is-perl), and other Unix languages | ASP and ASP.NET |

**Linux Server Overview**

A Linux server is a server based on the open-source Linux operating system. It is a popular choice due to its [low cost](https://phoenixnap.com/glossary/low-cost-it/), strong community support, and open-source code.

Many distributions are built around the Linux kernel – [Ubuntu](https://phoenixnap.com/kb/install-ubuntu-20-04), [CentOS](https://phoenixnap.com/kb/how-to-install-centos-7), [Debian](https://phoenixnap.com/kb/how-to-install-debian-10-buster), and others. Each offers an OS that is stable, secure, and flexible. Due to these features, Linux servers are widely used for hosting web services, network and system administration, database management, and other critical application requirements.

Although it offers distribution versions with a graphical user interface, Linux is mainly operated through a command line. Hence, it requires some knowledge about running operations through a terminal (or time to master doing so). Because it is primarily based on the command line, it offers excellent flexibility when interacting with the server. Additionally, it allows modifying configuration without rebooting the system.



This server option has no licensing fee. Unless you are the administrator, you will need to pay technical staff for installing and maintaining the server, but there is no additional cost for the server OS. Using open-source software is definitively cost-efficient but has the disadvantage of lacking official support. If you run into an issue, you will have to rely on the community of Linux users. While this popular OS solution has a large group of community supporters who are highly involved, you may have to dig deeper to find the answer for your specific use case.

System administrators have great flexibility and freedom with setting up the Linux server. This type of server has integrated remote administration and backend accessibility allowing admins the opportunity to optimize the system from anywhere.

Running a Linux server allows you to use and integrate other open-source software seamlessly. Although running Windows applications on Linux is possible, it requires additional software or even a VM machine. Linux servers support [MySQL](https://phoenixnap.com/kb/install-mysql-ubuntu-20-04)and [PostgreSQL](https://phoenixnap.com/kb/how-to-install-postgresql-on-ubuntu)databases and Unix programming languages (PHP, Perl, Python). So, if you are planning on utilizing these tools, Linux is the preferred option.

**Advantages of Linux Servers**

* No additional licensing fee as the operating system is free.
* More reliable - it rarely experiences malware, [cyber threats](https://phoenixnap.com/blog/what-is-cyber-security), or other security errors.
* Not demanding on the client hardware and lower resource consumption.
* Due to its low infrastructure requirements, it shows excellent performance rates.
* System administrators have the freedom and opportunity to customize the system.
* Seamless use of open-source software on the server.
* Supports cooperative work without exposing the program’s core.

**Disadvantages of Linux Servers**

* Operating via a command line instead of a GUI requires some learning or experience.
* Not all versions have [long-term support](https://phoenixnap.com/kb/centos-8-early-eol-in-2021).
* Updating from one major version to another can sometimes be complex.
* Some third-party and professional programs may not have support or require admin privileges.

Learn more about [Linux ransomware](https://phoenixnap.com/blog/linux-ransomware) attacks, what are the most famous ones and what you can do to protect your system.

**Linux Server Pricing**

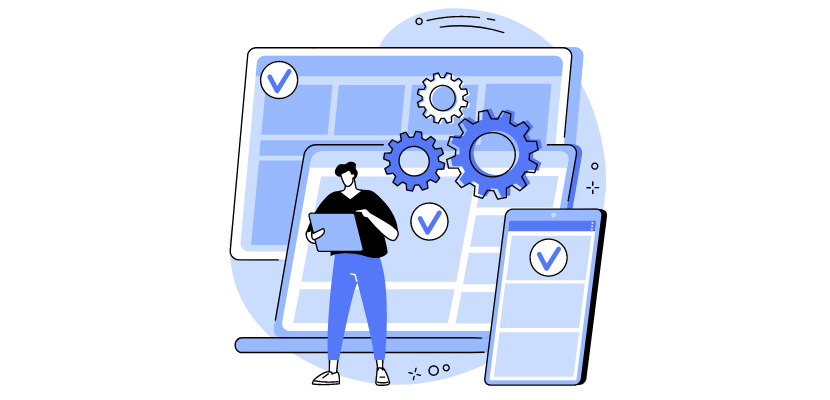
Linux is a lightweight OS that demands less hardware resources for handling more workload, which means it doesn’t usually have a high infrastructure cost. As mentioned above, there is no licensing for Linux distributions, except for Red Had and SUSE (platforms used for enterprise-level companies). An important cost to consider is having an in-house administrator competent to deal with Linux configurations and any potential issues.

**Windows Server Overview**

Windows Server is a commercial server solution created by Microsoft and powered by the Windows OS. They represent a popular beginner-friendly solution that is robust, reliable, and low-maintenance.

The term ‘Windows Server’ is a brand name encompassing Microsoft’s server operating systems, first released in 2003. The latest full release was in August 2021, when Windows Server 2022 was introduced.

Unlike Linux servers, Windows requires purchasing a license per user to use its services. Therefore, if you are renting a Windows server, the vendor will add the licensing fee to your overall bill for hosting and maintenance. The upside to using a paid OS is the long-term support it offers. Windows Servers have five years of maintenance plus five years of extended support from Microsoft.



Most beginners opt for Microsoft servers because of their user-friendly interface. Each release offers a complete GUI desktop experience that is very intuitive and polished. Additionally, this solution includes easy and optional automized system updates and the opportunity to solve technical issues through system recovery. Therefore, admins with less experience or time for maintenance find this out-of-the-box functionality very useful.

Windows Server can seamlessly integrate Windows applications and should be your first choice if you plan to use Microsoft SQL, Microsoft Access, or any other Microsoft program.

**Advantages of Windows Servers**

* Beginner-friendly due to its intuitive graphical user interface and out-of-the-box functionality.
* Guaranteed five years of maintenance + five years of extended support.
* Supports third-party applications and is compatible with Microsoft applications.
* Requires less admin monitoring and maintenance thanks to its robust approach and automated updates.

**Disadvantages of Windows Servers**

* Higher costs due to the obligatory licensing fee for the OS.
* More prone to malware, cyber-threats, and other security-related errors.
* Its mandatory GUI makes it more resource intensive.

**Windows Server Pricing**

Windows servers require more resources and, in most cases, have higher infrastructure costs compared to Linux. Also, they include a licensing fee which differs depending on the edition and vendor. The latest Windows Server 2022 Standard Edition (used for physical or minimal virtualized environments) starts from $1,069, whereas the Datacenter Edition starts from $6,155. For cloud configurations, the licensing price is usually paid year in and year out. When it comes to software support, Windows servers provide Microsoft Software Insurance which can be included in the purchase plan or paid as an additional add-on.

Refer to our [server cost guide](https://phoenixnap.com/kb/server-cost) to gain valuable server pricing insight for renting and building a server.

**Linux vs. Windows Server: Which One to Choose**

When deciding between a Linux server and a Windows server, bear in mind three important aspects:

* COST
* ADMIN EXPERIENCE
* YOUR SPECIFIC USE CASE

If you are searching for a more cost-efficient solution, Linux is undoubtedly the better option as there is no additional fee for running the operating system.

On the other hand, if the server administrator has no experience in [managing and maintaining](https://phoenixnap.com/blog/server-management) a server through a command line, you are better off paying the license for the Windows server. Not only will you have to spend less time monitoring and configuring the server, but you will also have access to Microsoft’s official support.

Finally, take into consideration what you want to run on the server. As Linux has PHP and MySQL support, installing WordPress is easier on a Linux server. Also, it offers easier access to HTTP, Apache, and other site-creation tools, JavaScript and NodeJS environments, and Perl and Python programming languages.

However, if you plan to develop web pages using a Microsoft framework, such as ASP or .NET frameworks, it is much simpler to use them on a Windows Server. The same applies to programs like Microsoft SQL, SharePoint, and Microsoft Access.

If you want to learn more about [Server operating systems](https://phoenixnap.com/kb/server-operating-system), read our article and find out what are the different types of Server OS and how to choose.

<https://phoenixnap.com/blog/linux-vs-microsoft-windows-servers>

# Difference Between OSI Model and TCP/IP Model

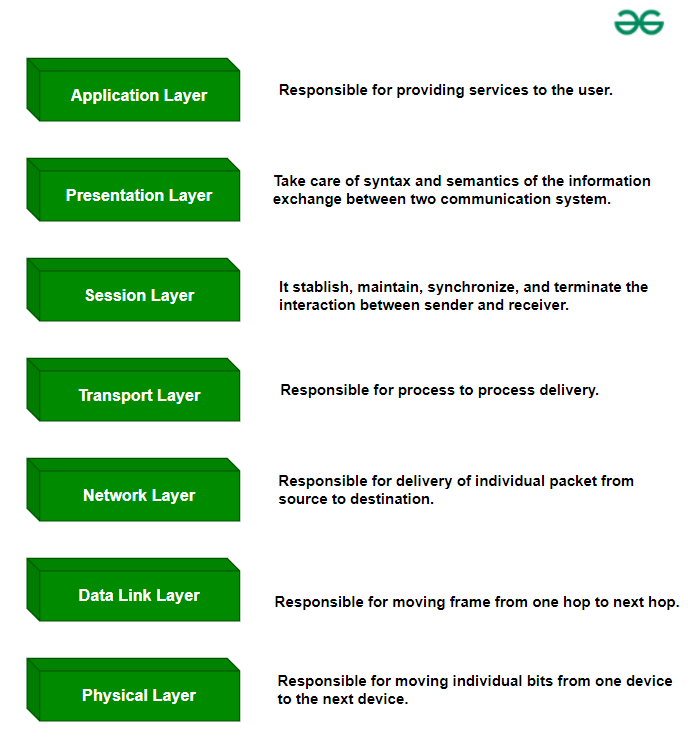
**Last Updated :**01 Aug, 2024

Data communication is a process or act in which we can send or receive data. Understanding the fundamental structures of networking is crucial for anyone working with computer systems and communication. For data communication two models are available, the OSI (Open Systems Interconnection) Model, and the TCP/IP (Transmission Control Protocol/Internet Protocol) Model.

These models work as frameworks for organizing and understanding how data moves from one device to another across networks. While both models aim to achieve similar goals, they differ in their approach, layer organization, and practical application within computer networking. We will discuss these two models in this article and also see the differences between the two models.

## OSI Model

[OSI](https://www.geeksforgeeks.org/layers-of-osi-model/)stands for Open Systems Interconnection. It has 7 layers [Physical layer](https://www.geeksforgeeks.org/physical-layer-in-osi-model/), [Data Link layer](https://www.geeksforgeeks.org/data-link-layer/), [Network layer](https://www.geeksforgeeks.org/network-layer-in-osi-model/), [Transport layer](https://www.geeksforgeeks.org/transport-layer-in-osi-model/), [Session layer](https://www.geeksforgeeks.org/session-layer-in-osi-model/), [Presentation layer](https://www.geeksforgeeks.org/presentation-layer-in-osi-model/), and [Application layer](https://www.geeksforgeeks.org/application-layer-in-osi-model/). Each layer performs its task independently. It was developed in 1984 by the International Organization for Standardization (ISO).

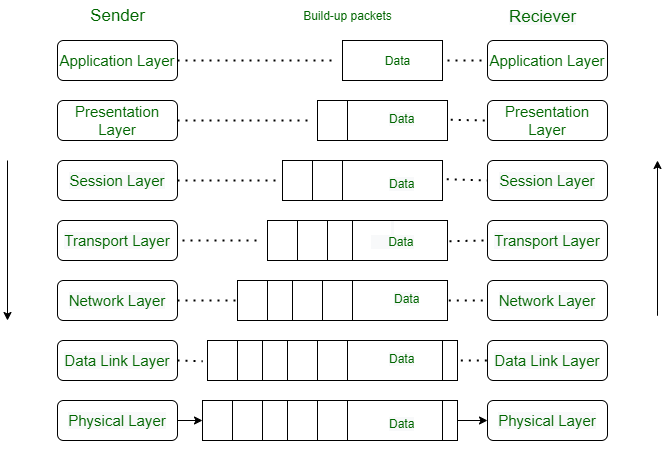


*OSI Model*

## Data Flow in OSI Model

The data flow in the OSI (Open Systems Interconnection) model describes how data is transmitted from one device to another through the seven layers of the OSI model. This process involves encapsulation and decapsulation at each layer to ensure proper data transmission and reception.

The data flow in the OSI model involves encapsulating data at each layer on the sender side, transmitting it over the network, and decapsulating it at each layer on the receiver side to ensure the data reaches its intended destination correctly and reliably.



*OSI Model*

### ****Advantages****

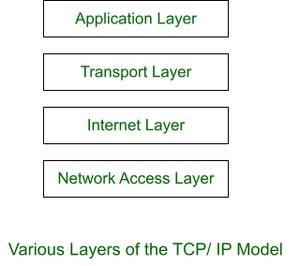
* Both connection-oriented services and connectionless services are supported.
* It is quite flexible.
* All the layers work independently.

### ****Disadvantages****

* Setting up a model is a challenging task.
* Sometimes, it becomes difficult to fit a new protocol into this model.
* It is only used as a reference model.

## TCP/IP Model

[TCP/IP](https://www.geeksforgeeks.org/tcp-ip-model/) stands for Transmission Control Protocol/Internet Protocol. It has 4 layers named as Physical layer, Network layer, Transport layer, and Application layer.  It also can be used as a communications protocol in a private computer network. It was designed by Vint Cerf and Bob Kahn in the 1970s.



### ****Advantages****

* Many Routing protocols are supported.
* It is highly scalable and uses a client-server architecture.
* It is lightweight.

### ****Disadvantages****

* Little difficult to set up.
* Delivery of packets is not guaranteed by the transport layer.
* Vulnerable to a synchronization attack.

## Similarities Between OSI Model and TCP/IP Model

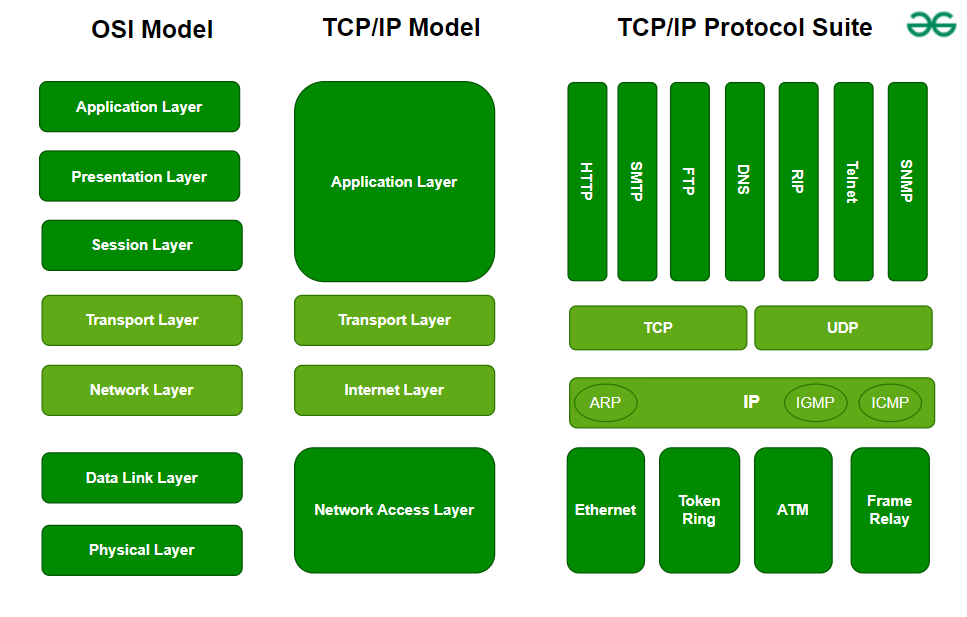
OSI and TCP/IP both are logical models. One of the main similarities between the OSI and TCP/IP models is that they both describe how information is transmitted between two devices across a network. Both models define a set of layers. Each layer performs a specific set of functions to enable the transmission of data.

Another similarity between the two models is that they both use the concept of [encapsulation](https://www.geeksforgeeks.org/encapsulation-in-java/), in which data is packaged into a series of headers and trailers that contain information about the data being transmitted and how it should be handled by the network.

For more information, you can refer [Similarities between TCP/IP model and the OSI model](https://www.geeksforgeeks.org/similarities-between-tcp-ip-model-and-osi-model/) article.

## Differences Between OSI Model and TCP/IP Model

The OSI (Open Systems Interconnection) Model and the TCP/IP (Transmission Control Protocol/Internet Protocol) Model are two frameworks used to understand how data moves through networks. While they both help in organizing network communication, they have distinct structures and purposes. Understanding these differences is essential for anyone learning about or working with computer networks.



*OSI vs TCP/IP*

| **Parameters** | **OSI Model** | **TCP/IP Model** |
| --- | --- | --- |
| **Full Form** | OSI stands for Open Systems Interconnection | TCP/IP stands for Transmission Control Protocol/Internet Protocol |
| **Layers** | It has 7 layers | It has 4 layers |
| **Usage** | It is low in usage | It is mostly used |
| **Approach** | It is vertically approached | It is horizontally approached |
| **Delivery** | Delivery of the package is guaranteed in OSI Model | Delivery of the package is not guaranteed in TCP/IP Model |
| **Replacement** | Replacement of tools and changes can easily be done in this model | Replacing the tools is not easy as it is in OSI Model |
| **Reliability** | It is less reliable than TCP/IP Model | It is more reliable than OSI Model |
| **Protocol Example** | Not tied to specific protocols, but examples include HTTP (Application), SSL/TLS (Presentation), TCP (Transport), IP (Network), Ethernet (Data Link) | HTTP, FTP, TCP, UDP, IP, Ethernet |
| **Error Handling** | Built into Data Link and Transport layers | Built into protocols like TCP |
| **Connection Orientation** | Both connection-oriented (TCP) and connectionless (UDP) protocols are covered at the Transport layer | TCP (connection-oriented), UDP (connectionless) |

## Conclusion

In conclusion, while both the OSI Model and TCP/IP Model are essential for understanding network communication, they differ in their structure and practical application. The OSI Model provides a theoretical framework with seven layers, emphasizing clear separation of functions, while the [TCP/IP Model](https://www.geeksforgeeks.org/tcp-ip-model/), with its four layers, reflects the protocols used on the internet today. Each model offers unique insights into how data is transmitted across networks, catering to different aspects of network design, management, and troubleshooting.

### Wired Equivalent Privacy (WEP)

WEP developed in 1997, was designed to secure wireless networks using encryption and access restriction. However, its reliance on the insecure RC4 encryption and shared key authentication made networks vulnerable to attack. While WEP initially provided encryption similar to wired networks, its flaws were widely exploited by hackers, making it obsolete.

The protocol’s discontinuation created more robust alternatives, such as WPA (Wi-Fi Protected Access). Despite its flaws, WEP’s simplicity and widespread adoption originally drew attention, but its inherent vulnerabilities eventually overshadowed its benefits, emphasizing the significance of constantly updating wireless security standards.

### Wi-Fi Protected Access (WPA)

WPA, launched in 2003, emerged as an effective successor to WEP, addressing its flaws. WPA uses the temporal key integrity protocol (TKIP) encryption to improve key management and integrity checks. It has two modes: WPA-Personal for home networks and WPA-Enterprise for enterprises that use RADIUS servers.

WPA’s 128-bit encryption provides enhanced protection over WEP’s weaker encryption standards; however, it’s still comparably weaker than WPA2 resulting in potential flaws and compatibility difficulties. Furthermore, adopting WPA may necessitate hardware modifications, providing a problem for users with older equipment.

### Wi-Fi Protected Access II (WPA2)

WPA2, released in 2004, is the most popular wireless security standard that uses the AES encryption technique to provide strong security. Its advantages over WPA include better administration and lower vulnerability to assaults. WPA2 is widely adopted as the industry standard, ensuring device interoperability.

However, vulnerabilities such as the key reinstallation attack (KRACK) constitute a security risk. While appropriate for most home networks, difficulties arise in enterprise settings where sophisticated attacks are more widespread. Furthermore, older gear without WPA2 compatibility may require upgrades. Despite these issues, WPA2 remains critical to wireless network security, but with ongoing attempts to address growing threats and weaknesses.

### Wi-Fi Protected Access III (WPA3)

WPA3, launched in 2018, provides greater encryption, protection against dictionary brute force attacks, and simpler device configuration via Wi-Fi Easy Connect. Despite these improvements, widespread acceptance is sluggish. WPA3 comes in three types: WPA3-Personal for home use, WPA3-Enterprise for organizational settings, and Wi-Fi Enhanced Open for non-password-protected networks.

While it enhances overall network security, drawbacks include deployment complexity, low user adoption, and compatibility issues with older devices and equipment. Despite its benefits, full-scale deployment of WPA3 has yet to occur, signaling a slow shift from older security protocols to this more modern standard.

### ****Security Implications of Wireless Networks****

Wireless networks, particularly Wi-Fi, offer unmatched convenience in providing internet access without the need for physical cables. However, this flexibility comes with a variety of security risks. Below are the most common security implications associated with wireless networks:

#### 1. **Eavesdropping and Interception**

Wireless networks transmit data over radio waves, which means that anyone within range of the network's signal can potentially intercept the communication. This makes sensitive information, such as passwords, personal data, or financial details, vulnerable to unauthorized access. Without encryption, attackers can easily capture and read unprotected network traffic, posing a significant risk.

#### 2. **Man-in-the-Middle (MITM) Attacks**

In a MITM attack, an attacker intercepts and potentially alters the communication between two parties who believe they are directly communicating with each other. On wireless networks, this could happen if an attacker positions themselves between a device and the access point (router). They could inject malicious data, manipulate messages, or steal sensitive information such as login credentials.

#### 3. **Rogue Access Points (Evil Twin)**

A rogue access point, also known as an "Evil Twin," is an unauthorized access point set up to impersonate a legitimate one. Attackers may deploy these devices, often using the same name (SSID) as a legitimate Wi-Fi network, in an attempt to trick users into connecting. Once connected, all data transmitted through the rogue access point can be captured, or the attacker might inject malicious traffic into the communication.

#### 4. **Denial of Service (DoS) and Jamming Attacks**

Wireless networks are susceptible to jamming attacks where the attacker sends noise or other interference on the same frequency, effectively disrupting communication and making the network unusable. Denial-of-Service (DoS) attacks can also be launched against wireless networks, where an attacker floods the network with unnecessary data to overwhelm it and prevent legitimate access.

#### 5. **Unauthorized Access**

If a wireless network is not properly secured, unauthorized users can gain access to the network. This can lead to various security risks, such as the theft of bandwidth, unauthorized access to shared files and resources, or the execution of malicious activities, such as launching further attacks on connected devices.

#### 6. **Weak or Default Passwords**

Many wireless networks, especially in home settings, use weak passwords or default credentials provided by the router manufacturer. Attackers can easily exploit this by using brute-force methods or password-cracking tools to gain unauthorized access to the network. Once in, attackers can exploit vulnerabilities or intercept sensitive data.

### ****Security Protocols to Mitigate Risks****

Several security protocols have been developed to protect wireless networks from the above risks. The following protocols are commonly used to secure wireless communication:

#### 1. **Wired Equivalent Privacy (WEP)**

WEP was one of the first security protocols designed for wireless networks, aiming to provide the same level of security as a wired network by encrypting data transmitted over the air. However, WEP is now considered outdated and insecure due to its weak encryption and poor key management.

**Weaknesses of WEP:**

* Uses a weak encryption algorithm (RC4) that can be easily cracked.
* Vulnerabilities in the way encryption keys are managed and rotated, making it susceptible to attacks like the "fluhrer attack" and "keystream reuse."

**Current Recommendation:**

* WEP should be avoided due to its inherent vulnerabilities. More secure alternatives like WPA2 or WPA3 should be used instead.

#### 2. **Wi-Fi Protected Access (WPA) and WPA2**

WPA was introduced as a replacement for WEP, offering stronger encryption and improved key management. WPA2, the more secure successor of WPA, became the industry standard for Wi-Fi security.

* **WPA** uses the Temporal Key Integrity Protocol (TKIP) for encryption, which improved upon WEP but still had weaknesses.
* **WPA2**, which is currently the most widely used standard, uses Advanced Encryption Standard (AES) for encryption, offering a much higher level of security compared to WPA and WEP.

**WPA2 Features:**

* Strong encryption through AES, which is far more secure than the RC4 used in WPA.
* Dynamic encryption key generation, ensuring that each session uses unique keys, making it harder for attackers to crack the encryption.

**WPA/WPA2-PSK (Pre-Shared Key):**

* This is used for home and small office networks, where a shared key (password) is used to authenticate devices. However, its security is only as strong as the password used. If the password is weak or easily guessable, attackers can easily gain access.

**WPA2-Enterprise:**

* This version is intended for larger organizations and provides additional security by using an external authentication server (RADIUS) for device authentication. This ensures that each device is properly authenticated before being allowed to connect to the network.

#### 3. **Wi-Fi Protected Access 3 (WPA3)**

WPA3 is the latest and most secure version of Wi-Fi security, introduced to address the weaknesses of WPA2. It offers several enhancements:

**Key Features of WPA3:**

* **Simultaneous Authentication of Equals (SAE):** This is a more secure method of establishing a shared key between devices during the handshake process. SAE replaces WPA2's Pre-Shared Key (PSK) and provides better resistance against offline dictionary attacks.
* **Forward Secrecy:** Even if an attacker is able to intercept and decrypt a session’s traffic, they cannot decrypt past sessions. This ensures the confidentiality of past communication, even if the encryption key is compromised later.
* **Protected Management Frames (PMF):** Provides protection against attacks on management frames, such as deauthentication or disassociation attacks, which are often used in attacks like the Evil Twin or denial-of-service attacks.

**Security Benefits of WPA3:**

* Stronger encryption and key management, making it more difficult for attackers to crack the security.
* Enhanced protection against brute-force and dictionary attacks.
* Better privacy and security during the connection process.

#### 4. **802.1X Authentication**

**802.1X** is an IEEE standard used for network access control, particularly in enterprise environments. It is often used in conjunction with WPA2-Enterprise and WPA3-Enterprise to provide strong authentication before allowing devices to connect to the network.

* 802.1X uses a RADIUS (Remote Authentication Dial-In User Service) server to authenticate devices and users attempting to access the network.
* Each device must provide a valid username and password (or other forms of authentication, such as certificates or smart cards) before being allowed to join the network.

**Benefits of 802.1X:**

* Provides strong, centralized control over who can access the network.
* Better protection against unauthorized access since each device must go through a formal authentication process.
* Enables network segmentation based on user roles or device types.

#### 5. **Virtual Private Network (VPN)**

While VPNs are not a wireless security protocol by themselves, they are frequently used to secure data transmitted over unsecured wireless networks. VPNs encrypt all network traffic between a device and a remote server, ensuring that even if someone intercepts the data, they won’t be able to read it.

**Benefits of VPNs:**

* Provides end-to-end encryption, safeguarding sensitive data from eavesdropping.
* Useful for securing communication over public or untrusted networks, such as public Wi-Fi hotspots.

#### 6. **MAC Address Filtering**

MAC address filtering is a technique where only devices with specific MAC addresses (unique hardware identifiers) are allowed to connect to the network. While this adds an additional layer of security, it’s not foolproof since attackers can easily spoof MAC addresses to bypass this filter.

**Benefits of MAC Filtering:**

* Adds a layer of security by restricting access to only known devices.
* Helps reduce the risk of unauthorized devices connecting to the network.

**Limitations:**

* It is relatively weak, as MAC addresses can be spoofed by attackers with the right tools.

#### 7. **Security Monitoring and Intrusion Detection Systems (IDS)**

Security monitoring tools and Intrusion Detection Systems (IDS) are used to detect suspicious activity or attacks on wireless networks. These systems can identify unauthorized access attempts, rogue access points, or abnormal traffic patterns, and alert network administrators to potential security threats.

### ****Best Practices for Securing Wireless Networks****

To maximize the security of wireless networks, follow these best practices:

1. **Use Strong Encryption**: Ensure that your network uses WPA2 or WPA3 with AES encryption. Avoid outdated protocols like WEP.
2. **Enable WPA3**: If possible, upgrade to WPA3 for enhanced security, especially if you are concerned about the latest threats like brute-force attacks and eavesdropping.
3. **Use Strong, Unique Passwords**: Avoid using easily guessable passwords. Implement complex passwords and consider using a password manager for stronger security.
4. **Use VPNs**: For sensitive communications, especially when on public or unsecured networks, use a VPN to secure all traffic.
5. **Monitor the Network**: Regularly scan for rogue access points and monitor network traffic for unusual activity or unauthorized access attempts.
6. **Disable WPS**: Wi-Fi Protected Setup (WPS) can be vulnerable to brute-force attacks, so it’s best to disable it unless necessary.
7. **Regularly Update Firmware**: Keep routers and other networking hardware up to date with the latest security patches to fix known vulnerabilities.

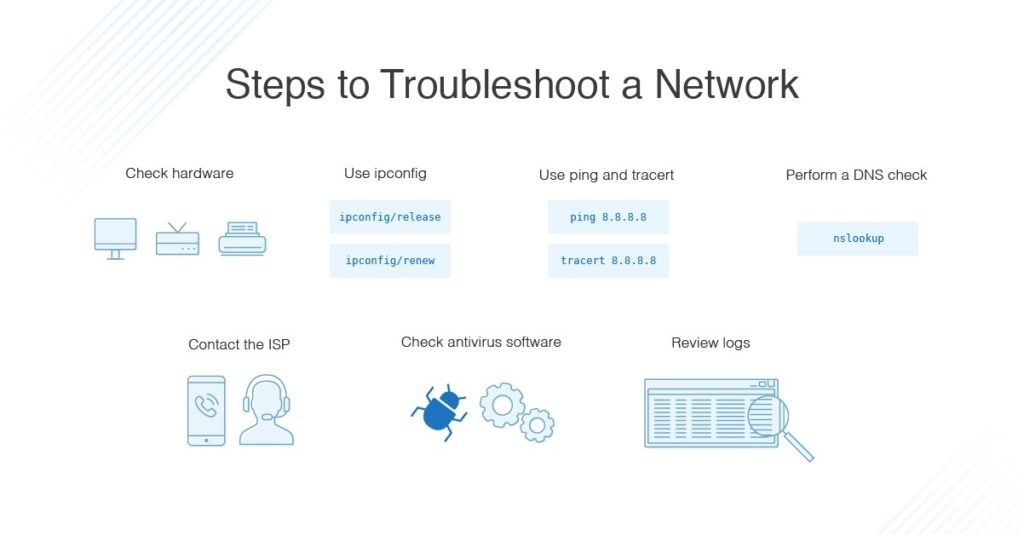
## Network Troubleshooting Steps

Issues and problems can arise at numerous points along the network. Before you start trying to troubleshoot any issue, you want to have a clear understanding of what the problem is, how it has arisen, who it’s affecting, and how long it has been going on. By gathering the right information and clarifying the problem, you’ll have a much better chance of resolving the issue quickly, without wasting time trying unnecessary fixes. You can always start by working through these simple network troubleshooting steps to diagnose the issue.

### 1. Check the hardware.

When you’re beginning the troubleshooting process, check all your hardware to make sure it’s connected properly, turned on, and working. If a cord has come loose or somebody has switched off an important router, this could be the problem behind your networking issues. There’s no point in going through the process of troubleshooting network issues if all you need to do is plug a cord in. Make sure all switches are in the correct positions and haven’t been bumped accidentally.

Next, turn the hardware off and back on again. This is the mainstay of IT troubleshooting, and while it might sound simplistic, often it really does solve the problem. Power cycling your modem, router, and PC can solve simple issues—just be sure to leave each device off for at least 60 seconds before you turn it back on.



### 2. Use ipconfig.

Open the command prompt and type “ipconfig” (without the quotes) into the terminal. The Default Gateway (listed last) is your router’s IP. Your computer’s IP address is the number next to “IP Address.” If your computer’s IP address starts with 169, the computer is not receiving a valid IP address. If it starts with anything other than 169, your computer is being allocated a valid IP address from your router.

Try typing in “ipconfig /release” followed by “ipconfig /renew” to get rid of your current IP address and request a new one. This will in some cases solve the problem. If you still can’t get a valid IP from your router, try plugging your computer straight into the modem using an ethernet cable. If it works, the problem lies with the router.

### 3. Use ping and tracert.

If your router is working fine, and you have an IP address starting with something other than 169, the problem’s most likely located between your router and the internet. At this point, it’s time to use the ping tool. Try sending a ping to a well-known, large server, such as Google, to see if it can connect with your router. You can ping Google DNS servers by opening the command prompt and typing “ping 8.8.8.8”; you can also add “-t” to the end (ping 8.8.8.8 -t) to get it to keep pinging the servers while you troubleshoot. If the pings fail to send, the command prompt will return basic information about the issue.

You can use the tracert command to do the same thing, by typing “tracert 8.8.8.8”; this will show you each step, or “hop,” between your router and the Google DNS servers. You can see where along the pathway the error is arising. If the error comes up early along the pathway, the issue is more likely somewhere in your local network.

### 4. Perform a DNS check.

Use the command “nslookup” to determine whether there’s a problem with the server you’re trying to connect to. If you perform a DNS check on, for example, google.com and receive results such as “Timed Out,” “Server Failure,” “Refused,” “No Response from Server,” or “Network Is Unreachable,” it may indicate the problem originates in the DNS server for your destination. (You can also use nslookup to check your own DNS server.)

### 5. Contact the ISP.

If all of the above turn up no problems, try contacting your internet service provider to see if they’re having issues. You can also look up outage maps and related information on a smartphone to see if others in your area are having the same problem.

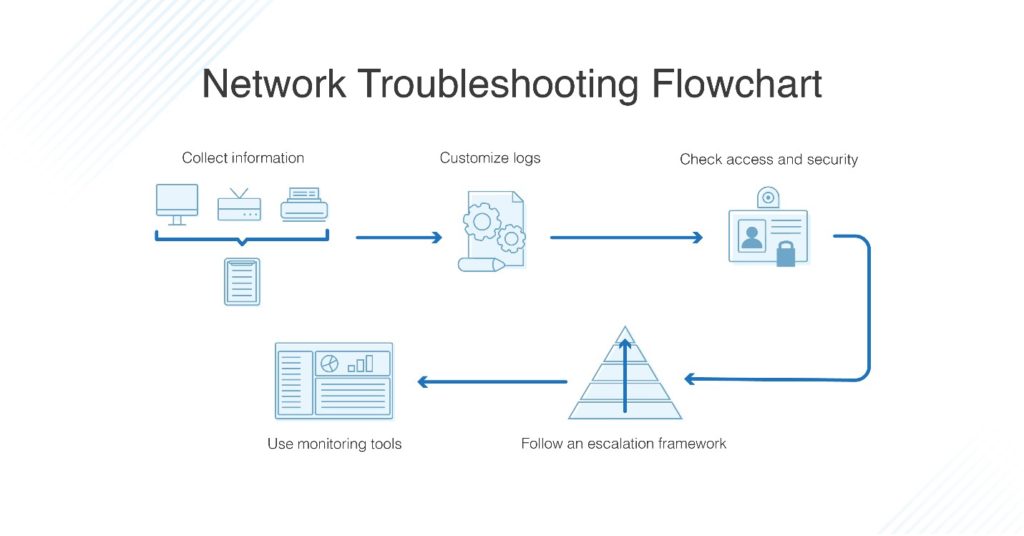
### 6. Check on virus and malware protection.

Next, make sure your virus and malware tools are running correctly, and they haven’t flagged anything that could be affecting part of your network and stopping it from functioning.

### 7. Review database logs.

Review all your database logs to make sure the databases are functioning as expected. If your network is working but your database is full or malfunctioning, it could be causing problems that flow on and affect your network performance.

## Network Troubleshooting Best Practices



To make troubleshooting as efficient as possible, it’s very important to have best practices in place. As you work through the steps to try to solve network issues, following these network troubleshooting best practices can help streamline the process and avoid unnecessary or redundant efforts.

### 1. Collect information.

To best support your end users, you first need to make sure you’re clear on what the problem is. Collect enough information from both the people who are experiencing network issues and the network itself, so you can replicate or diagnose the problem. Take care not to mistake symptoms for the root cause, as what initially looks like the problem could be part of a larger issue.

### 2. Customize logs.

Make sure your event and security logs are customized to provide you with information to support your troubleshooting efforts. Each log should have a clear description of which items or events are being logged, the date and time, and information on the source of the log (MAC or IP address).

### 3. Check access and security.

Ensure no access or security issues have come up by checking all access permissions are as they should be, and nobody has accidentally altered a sensitive part of the network they weren’t supposed to be able to touch. Check all firewalls, antivirus software, and malware software to ensure they’re working correctly, and no security issues are affecting your users’ ability to work.

### 4. Follow an escalation framework.

There’s nothing worse than going to the IT help desk and being directed to another person, who then directs you to another person, who directs you to yet another. Have a clear escalation framework of who is responsible for which issues, including the final person in the chain who can be approached for resolution. All your end users should know who they can go to about a given issue, so time isn’t wasted talking to five different people who cannot fix the problem.

### 5. Use monitoring tools.

Troubleshooting can be done manually but can become time-consuming if you go through each step. When you have a bunch of people knocking on your office door or sending you frantic emails, it can be overwhelming to try to find the problem, let alone fix it. In business and enterprise situations, it’s best to use monitoring tools to make sure you’re getting all the relevant network information and aren’t missing anything vital, not to mention avoiding exposing the company to unnecessary risk.

My preferred monitoring software is [SolarWinds® Network Performance Monitor (NPM)](https://www.solarwinds.com/network-performance-monitor?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-NPM-20190923_NetworkTroubles_X_X_VidNo_X-X). It’s a well-designed tool with features to support network troubleshooting issues in an efficient and thorough way. It allows you to clearly baseline your network behavior, so you have good data on what your network should look like and how it usually performs, and it includes advanced alerting features so you don’t receive floods of alerts all the time. You can customize the software to alert you to major issues, choose the timing of alerts, and define the conditions under which alerts occur.

Troubleshooting network connectivity issues is often a complex process, as there can be multiple layers of potential problems, ranging from physical connectivity issues to configuration problems or network congestion. Protocol analysis tools, such as **Wireshark**, **tcpdump**, or other network sniffers, can help diagnose and resolve these issues by capturing and analyzing network traffic in real-time. Below are the key steps involved in troubleshooting a network connectivity issue using protocol analysis tools:

**Step 1: Identify the Problem**

Before diving into the analysis, it's essential to clearly define the problem you're experiencing. This could include:

* **Symptoms**: Slow network performance, intermittent connectivity, inability to reach specific services, or complete lack of connectivity.
* **Scope**: Determine if the issue is isolated to a single device, a particular segment of the network, or the entire network.
* **Impact**: Identify which applications, services, or users are affected by the issue.

Understanding the problem will guide the data collection and analysis process.

**Step 2: Gather Information and Prepare the Environment**

Before starting protocol analysis, gather the following details:

* **Affected device(s)**: Note the IP address, MAC address, and the operating system of the device(s) facing the issue.
* **Network topology**: Understand the layout of the network, including routers, switches, firewalls, and any other devices in the path between the client and the server.
* **Configuration settings**: Check the configuration of network devices involved, such as DNS settings, IP addresses, subnet masks, and gateway settings.

Prepare the protocol analysis tool by installing and configuring it properly:

* **Wireshark**: One of the most widely used protocol analysis tools. Make sure it’s installed on a system that is connected to the network where the issue is occurring.
* **Filter settings**: Set up capture filters to focus on specific protocols (e.g., TCP, DNS, DHCP) or specific IP addresses to narrow down the traffic that needs to be analyzed.
* **Promiscuous mode**: Ensure the network interface is set to promiscuous mode to capture all packets, not just those destined for the local machine.

**Step 3: Capture Network Traffic**

Use the protocol analysis tool to start capturing network traffic:

* **Start the capture**: Begin monitoring traffic on the network. Choose the right network interface (e.g., Wi-Fi or Ethernet).
* **Reproduce the issue**: Try to reproduce the network problem (e.g., access a slow or unreachable website) to capture the relevant traffic during the problem occurrence.
* **Capture duration**: Depending on the nature of the issue, capture traffic for a few minutes or longer to ensure that you have enough data to analyze the problem.

**Step 4: Analyze Captured Data**

Once you have captured the network traffic, it's time to analyze the data to identify the cause of the problem. This step involves looking at the specific packets exchanged between devices:

* **Check for packet loss**: Look for signs of packet loss (such as repeated retransmissions) or latency issues in the captured traffic. Packet loss can indicate issues such as network congestion or faulty hardware (e.g., routers or cables).
* **Examine error messages**: Identify any error messages in the captured packets. For instance, TCP packets with the “RST” (Reset) flag indicate a connection problem, while ICMP packets with a "Destination Unreachable" message can point to routing or DNS issues.
* **Analyze protocol behavior**: Look at specific protocols:
  + **ARP (Address Resolution Protocol)**: Check if there are issues resolving MAC addresses for IP addresses.
  + **DNS (Domain Name System)**: If the problem is related to name resolution, look at DNS request and response packets to check if DNS servers are responding properly.
  + **DHCP (Dynamic Host Configuration Protocol)**: If IP address assignment is an issue, analyze DHCP Discover and Offer packets to ensure the client is receiving a valid IP address.
  + **TCP**: Look for issues like slow handshakes, repeated retransmissions, or connection resets (RST) that might indicate network congestion, firewall blocks, or other issues with TCP connections.

**Step 5: Identify the Root Cause**

Based on your analysis of the traffic, identify the root cause of the connectivity issue:

* **Network Congestion**: If you see high latency or dropped packets, network congestion might be the issue. Look for signs of buffer overflows in routers or switches.
* **Routing Issues**: If packets are not reaching their destination or taking long paths, you might have routing problems. Look for missing or incorrect routing tables.
* **DNS Issues**: If DNS queries are timing out or returning incorrect IP addresses, the problem might lie with the DNS server.
* **Authentication Problems**: If the issue is with accessing a secured service, check for failed authentication attempts or security-related errors.
* **Hardware Failures**: If you notice consistent packet drops, retransmissions, or network unreachability, the problem could be due to faulty hardware (e.g., network interface cards, routers, or switches).

**Step 6: Resolve the Issue**

After identifying the root cause, take the necessary steps to resolve the issue:

* **Network Configuration**: Correct any misconfigurations, such as IP address conflicts, incorrect DNS settings, or wrong gateway settings.
* **Replace Faulty Hardware**: If a physical device (e.g., network cable, switch, router) is identified as the issue, replace or repair the faulty hardware.
* **Adjust Network Load**: If congestion is the problem, consider upgrading network hardware, optimizing traffic, or implementing Quality of Service (QoS) rules to prioritize important traffic.
* **Firewall and Security Settings**: If security or firewall rules are blocking traffic, modify or adjust firewall rules to allow the necessary traffic while maintaining security.

**Step 7: Test the Solution**

After implementing the fix, test the network connectivity again to ensure the issue is resolved:

* **Reproduce the problem**: Try to access the same service or application that was causing the issue.
* **Monitor network traffic**: Use the protocol analyzer to capture new traffic and confirm that the issue is no longer present. Check that packets are no longer being dropped, and connections are established successfully.
* **Verify performance**: Ensure that the network is operating at expected performance levels without unnecessary delays or errors.

**Step 8: Document the Findings**

Once the issue is resolved, document the findings, including:

* **Description of the issue**: A clear summary of the symptoms, affected devices, and network components.
* **Root cause**: A detailed explanation of what caused the issue.
* **Solution**: The steps taken to resolve the issue and any changes made to the network configuration or hardware.
* **Lessons learned**: Any important insights gained that could prevent future issues or help with troubleshooting similar problems.

Routing protocols are used to automatically and dynamically exchange routing information between routers. There are several routing protocols to choose from, each with its own pros and cons, as each routing protocol is designed to be well suited to a particular network implementation scenario. Two of the most popular routing protocols used today are Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). These are very different in their design, as we shall see. We’ll start with a summarized version of the differences and then explain each protocol separately in more depth.

### OSPF

OSPF is an interior gateway protocol (IGP) that can route packets inside a single [Autonomous System](https://en.wikipedia.org/wiki/Autonomous_system_(Internet)) (AS). Unlike other IGPs, OSPF is a link-state routing protocol. In other words, it relies on link-state information to calculate route paths and make routing decisions.

After the protocol starts, each router running OSPF sends link-state advertisements (LSAs) throughout the AS or area containing information about its connected interfaces and routing metrics. When there is a change to any of the routers, the change is propagated to all the routers in the area. Such an update triggers a rerun of the shortest-path-first algorithm.

OSPF splits each AS into smaller sections called areas. All the routers in the same area have identical LSA databases. They also have summarized information about the other areas. There are multiple types of OSPF areas, which will be described later in this article.

### BGP

[BGP](https://www.catchpoint.com/bgp) is a routing protocol primarily used to perform inter-domain routing, and is considered an External Gateway Protocol (EGP).  However, BGP can also be used to advertise networks within an AS, and when configured to do so, can also function in a similar manner to IGPs.

BGP is used to exchange routing information among routers in the same AS or different ASs. An AS is a set of routers under a single administrative authority. An AS path is the route to a destination. It is also a list of ASs that the route passes through to reach a particular router. Each route has additional information attached that comes in the form of path attributes. The path attributes are used in routing policies to influence how the router routes the traffic.

### Summary of differences between OSPF and BGP

Below is a summary of some of the differences between OSPF and BGP.

|  |  |  |
| --- | --- | --- |
| **Feature** | **OSPF** | **BGP** |
| Routing domain | Intra-domain | Primarily Inter-domain but can also be used as intra-domain |
| Maximum size of the network | Can be deployed in mid-sized to large networks with up to several hundred routers | Scalable to the worldwide Internet |
| Implementation | Easy for basic configuration | Easy to moderately difficult for basic configuration |
| Network topology | Hierarchical | Mesh, but can be modified to star using a route reflector |
| Convergence | Fast | Slow |
| Resource requirements | Memory- and CPU-intensive | Directly proportional to the size of the routing table |
| Metric | Based on bandwidth/cost | Based on BGP attributes |

## Differences between OSPF and BGP

There are a number of differences between OSPF and BGP. To start with, OSPF is an interior gateway protocol. Therefore, it is confined to a single domain for routing (intra-domain). On the other hand, BGP is primarily designed to be used to route between routing domains (inter-domain).

OSPF can be successfully deployed in networks with several hundred routers in a single flat area. However, this is in direct relation to the resources available on the routers (read below about the resource requirements). Conversely, the only routing protocol that runs on the Internet exclusively is BGP.

Basic configuration of OSPF (say a single area with no fancy features deployed) is relatively easy. Even the most basic BGP configuration requires more effort than OSPF’s basic configuration (and some advanced routing knowledge). While both OSPF and BGP can get very complex, BGP is far more difficult to use due to numerous available features that make it suitable in many situations and corner cases.  For example, OSPF primarily focuses on the metric to determine the best route. BGP on the other hand uses a series of attributes that can be adjusted very granularly, to modify routing behavior in multiple ways.

OSPF must be deployed hierarchically (we will discuss this in the next section), whereas BGP does not require any hierarchy to scale.

In terms of convergence, OSPF reacts faster to network changes than BGP. This makes sense, given that BGP is designed for vast networks where changes happen more often statistically. You would not want routers on large networks to be constantly recalculating routes.

As for resource requirements, because OSPF requires constant calculation, it is considered CPU- and memory-intensive. In contrast, BGP does not react that quickly but becomes CPU- and memory-intensive when the size of the routing table increases. Therefore, routers holding the Internet routing table require powerful CPUs and lots of memory.

Regarding the metric used to calculate the best route, OSPF uses cost, derived from interface bandwidth plus the cost advertised by the other routers when the LSA is sent. BGP can use any BGP attributes to select the best route.

## The inner workings of OSPF and BGP

This section describes the two protocols in more detail. We will focus on the most common options that we will see configured in the article’s next section.

### OSPF

As mentioned in the first section, OSPF is a link-state routing protocol. OSPF routers exchange link-state advertisements that describe networks they know. From these advertisements, each OSPF router builds within its memory a full topology of the network.  Before doing this, though, they need to establish an adjacency with neighboring OSPF routers. Before establishing an adjacency, the two routers must become neighbors. The routers find each other using Hello packets.

The following information from the Hello packets sent by the two routers must match:

* They should be in the same area.
* The router ID must be unique.
* The subnet must be the same.
* The Hello and dead timer must be the same.
* The stub flag must match.
* The authentication must match.

It’s important to know that not all neighbor routers become adjacent as well. Let’s consider the scenario in Figure 1 (a broadcast or a nonbroadcast multiaccess network).

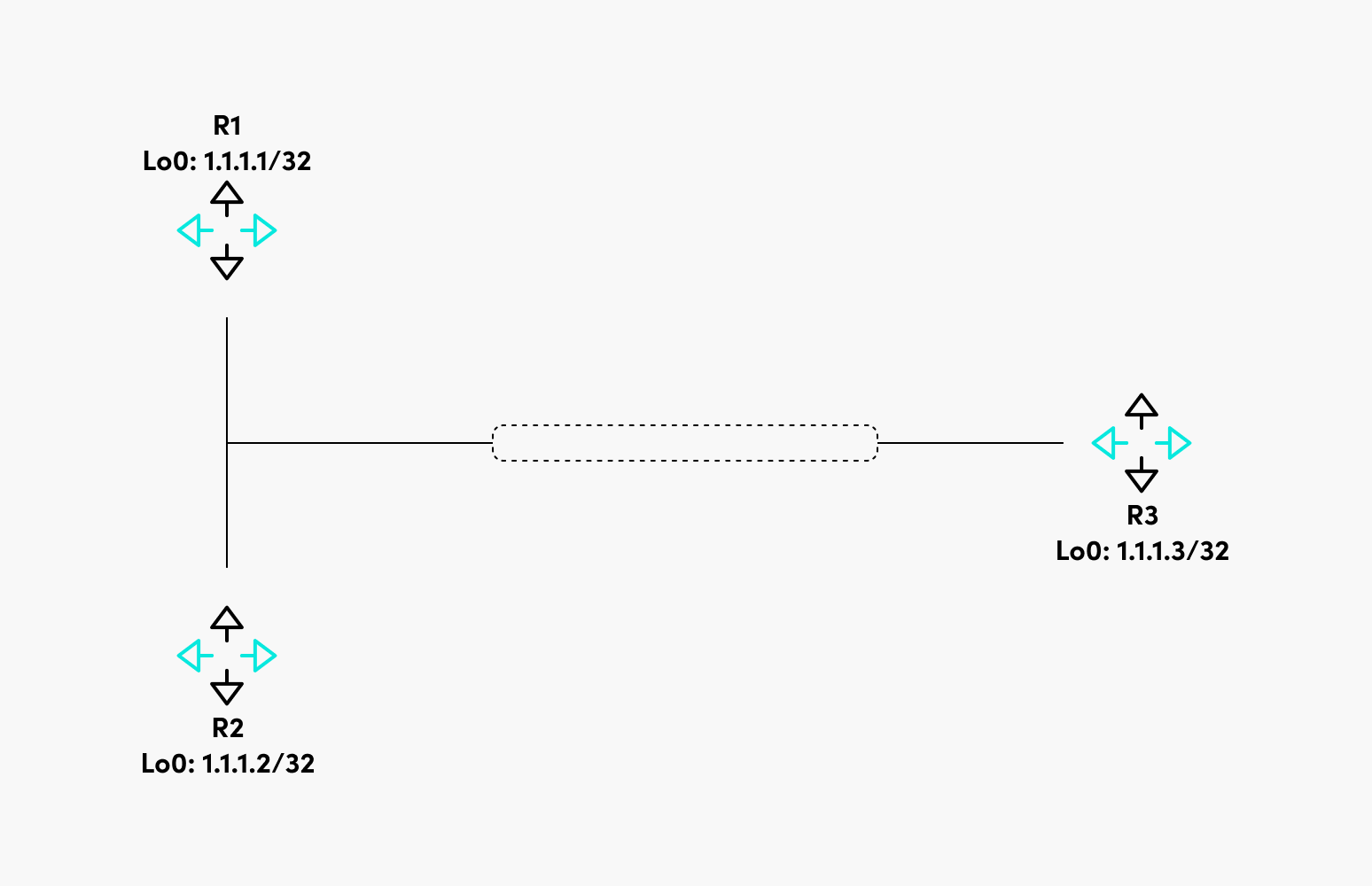


Figure 1: Broadcast multiaccess network.

The OSPF Hello protocol will elect a designated router (DR) for the network in this particular example. For redundancy purposes, a backup designated router (BDR) will be elected. Every other router from the segment will become a DROther. This means that the DROther will become adjacent only with the DR and BDR, and every DROther router will receive the LSA from the DR (or BDR, in case the DR fails).

The purpose of this mechanism is to reduce the amount of routing information traffic exchanged. There are two rules to elect the DR and BDR:

* **Priority:**Highest priority is preferred.
* **Router ID:**Highest router ID is preferred.

The router ID is derived using the following options:

* Manually set.
* The highest IP address from a loopback interface.
* The highest IP address from an up physical interface.

Should every OSPF configuration be left at default and all the routers be configured at exactly the exact moment, in the above case, R3 will become the DR, R2 the BDR, and R1 the DROther (Figure 2).

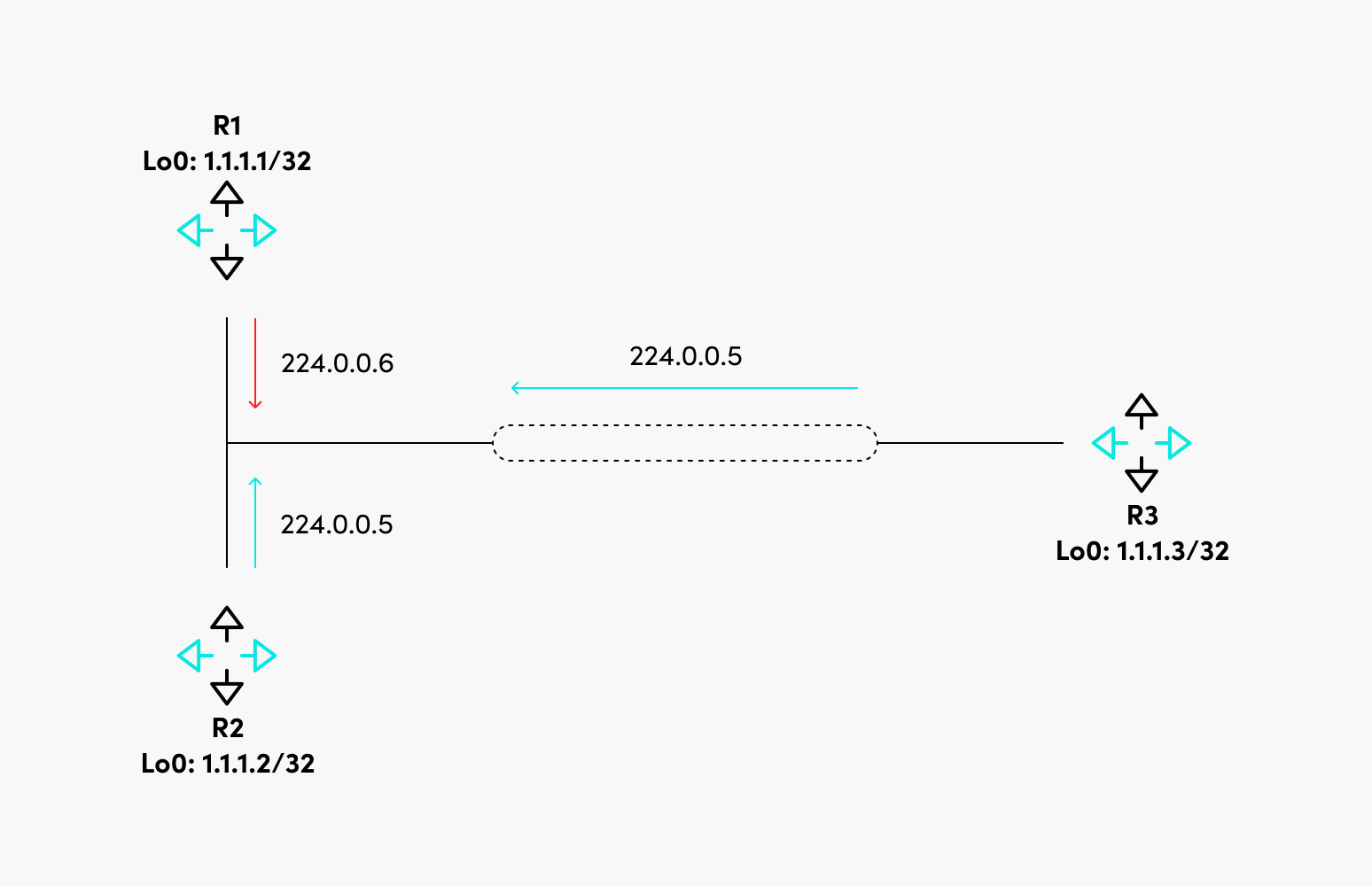


Figure 2: DR, BDR, and DROther.

Above, the DROther sends its updates to the multicast IP address 224.0.0.6, on which only the DR and BDR are listening. The DR sends the update to 224.0.0.5, to which all the routers from the segment are listening.

It was mentioned that OSPF was designed to be hierarchical to scale, which is achieved by using OSPF areas. Based on the type of LSAs that can be present in an area, these are the OSPF areas:

It is worthwhile to discuss what each LSA type is:

* **LSA Type 1 - Router LSA:** Generated by every router and describes the router links.
* **LSA Type 2 - Network LSA:** Generated by the DR and describes the routers connected to the segment.
* **LSA Type 3 - Network Summary LSA:** Generated by the Area Border Router and sent to another area to represent the destinations outside that area.
* **LSA Type 4 - ASBR Summary:** Used to describe the router that advertises external routes.
* **LSA Type 5 - AS External LSA:** Represents the routes external to the AS.
* **LSA Type 7 - NSSA External:** Represents the external routes from an NSSA area that will be converted to Type 5 LSA.

Now let’s discuss the router types in OSPF, as shown in Figure 3.

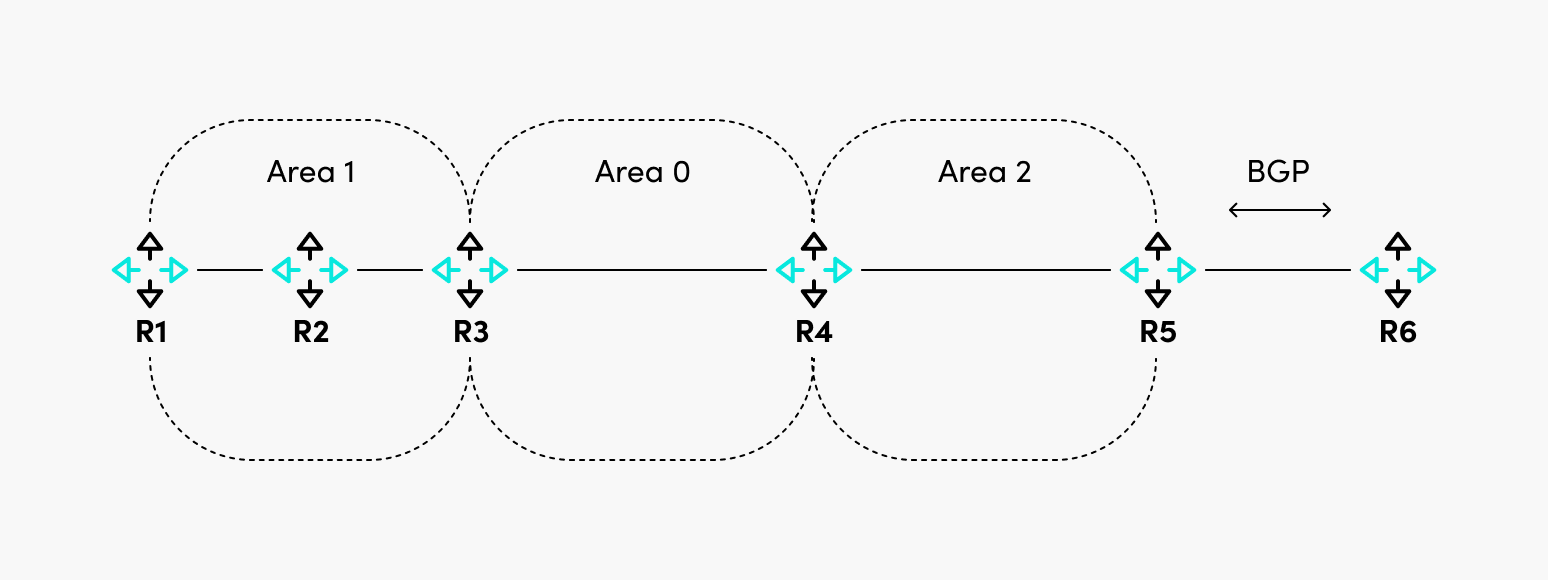


Figure 3: OSPF areas and router roles.

In this particular case, area 0 is the backbone area, the rest of the areas are non-backbone areas, and these are the roles of the routers:

* **R1 and R2:** Internal routers because all their interfaces are in the same area.
* **R3 and R4:** Area Border Routers because their interfaces are in two different areas. They are also called backbone routers because they have at least one interface in Area 0.
* **R5:** Autonomous System Border Router, which redistributes external routes (BGP) in OSPF.

### BGP

As already mentioned, BGP is used to connect ASes or to advertise network reachability information inside an AS.

When BGP is configured between routers in the same AS, it is called Internal BGP. When it is configured between routers in different ASes, it is called External BGP (Figure 4).

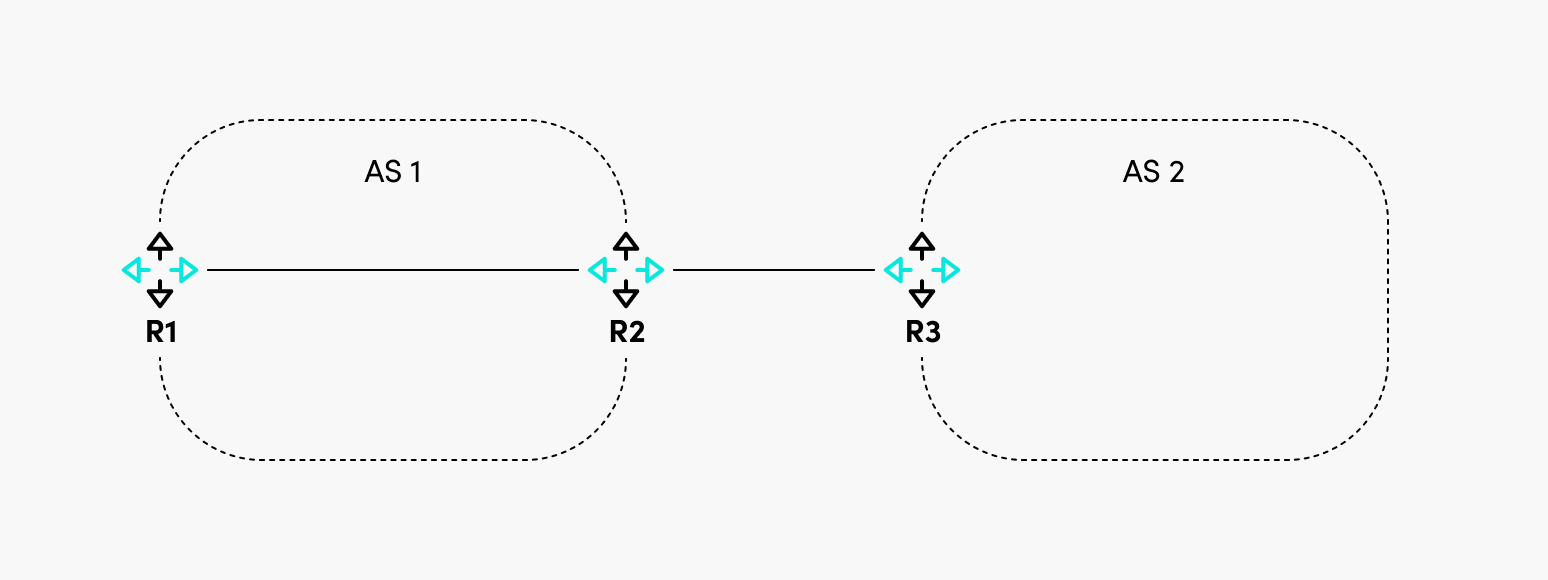


Figure 4: Internal and external BGP peers.

In this example, between R1 and R2, the BGP session is internal, and between R2 and R3, the BGP session is external. The network reachability information is sent through BGP Update messages, allowing the routes’ advertisement and withdrawal.

One of the most critical fields in the Update message is the “Path Attributes,” which define what attributes are attached to the routes. There are four categories of BGP attributes:

* **Well-Known Mandatory:** These are recognized by all BGP speakers and must be present in all Update messages.
* **Well-Known Discretionary:** All BGP speakers recognize these messages, but they can optionally be present in Update messages.
* **Optional Transitive:** These may or may not be recognized by BGP speakers, but even so, they are passed to other BGP peers.
* **Optional Non-Transitive:** The BGP speakers might recognize these messages but not pass them to other BGP peers.

While OSPF uses cost as a metric to determine the best path, BGP uses BGP attributes to determine the best path. Because it is not uncommon to have multiple paths to the same destination, BGP has a best-path selection algorithm to eventually choose the best path (or paths, if BGP multipath is configured).

Detailed information about the exact steps for the algorithm are described [here](https://www.catchpoint.com/network-admin-guide/bgp-attributes) and [here](https://www.catchpoint.com/network-admin-guide/bgp-communities).

One thing to remember is that a route will be considered suitable as a candidate for the best path only if the next hop to reach that route is reachable.

Consider Figure 5. By default, if no additional actions are taken on R2, R1 will not accept the 10.10.10.0/24 route because, in the BGP Update message, the next hop for the route is R3, and R1 does not know how to reach it.

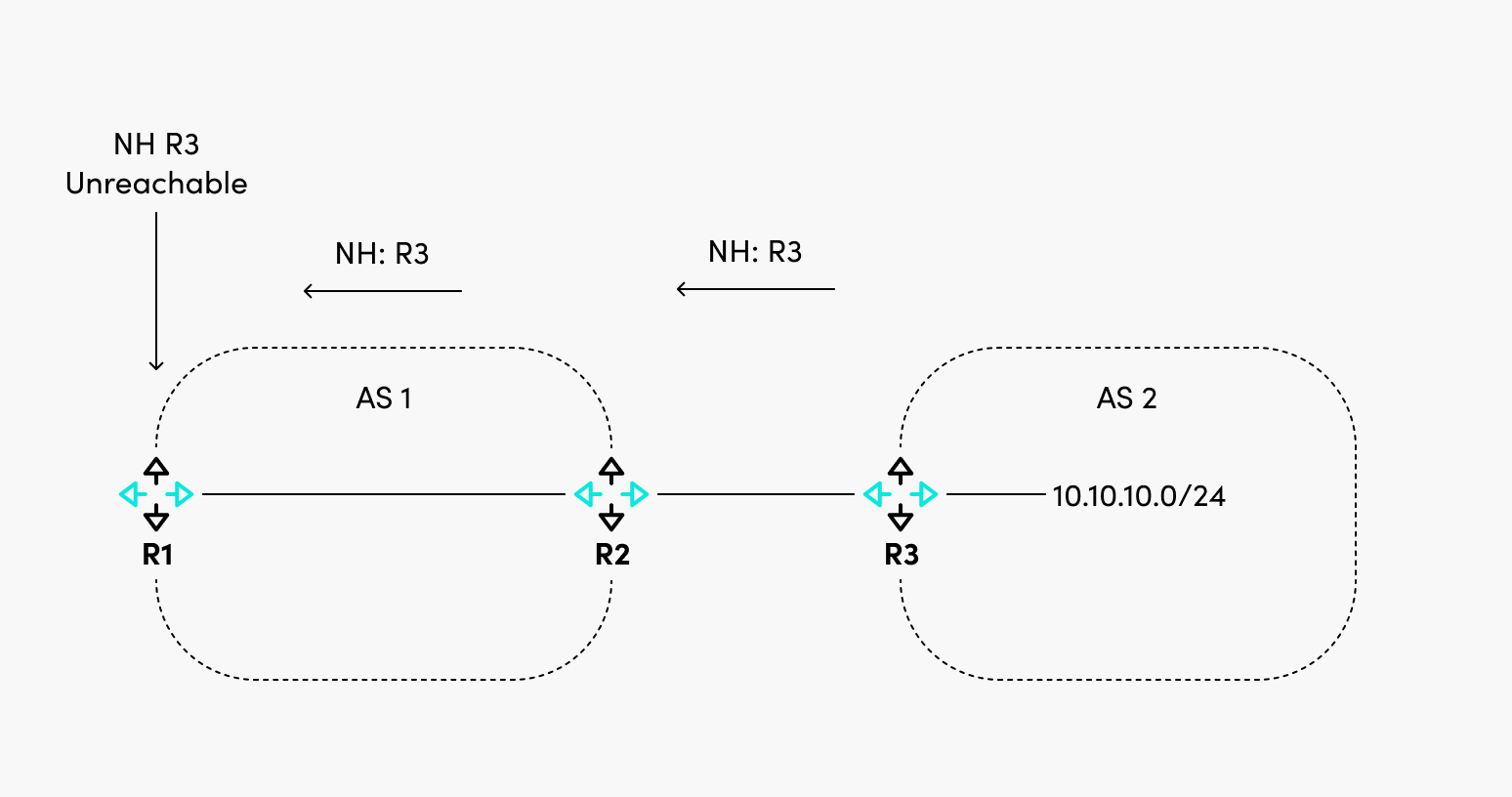


Figure 5: BGP route next hop must be reachable.

There are multiple ways to solve the problem above, and they are listed below:

* R2 can set itself as the next-hop-self for the BGP update sent to R1, so R1 now has the next-hop IP address within its routing table.
* R2 can advertise the subnet between R2 and R3 in the IGP of AS 1. This however, is less desirable, since inter AS communication should be restricted only to BGP, otherwise unpredictable routing may occur.  It is best practice to keep IGPs operating within the boundaries of their ASes.

Free product tour

#### How can IPM improve your network reachability?

Discover how to rapidly pinpoint network issues: take our product tour to see how we identified & resolved latency in just a few clicks!

[**Try it Out**](https://www.catchpoint.com/guided-product-tour-network)

## Configuration

This section shows how to configure basic OSPF and BGP on Cisco routers.

### OSPF

For OSPF, we will use the following diagram of a multi-area OSPF network. In this scenario, R3 is the ABR because it has interfaces in both Area 0 and Area 1. Area 1 is a non-backbone area (not a stub, stubby, or NSSA).

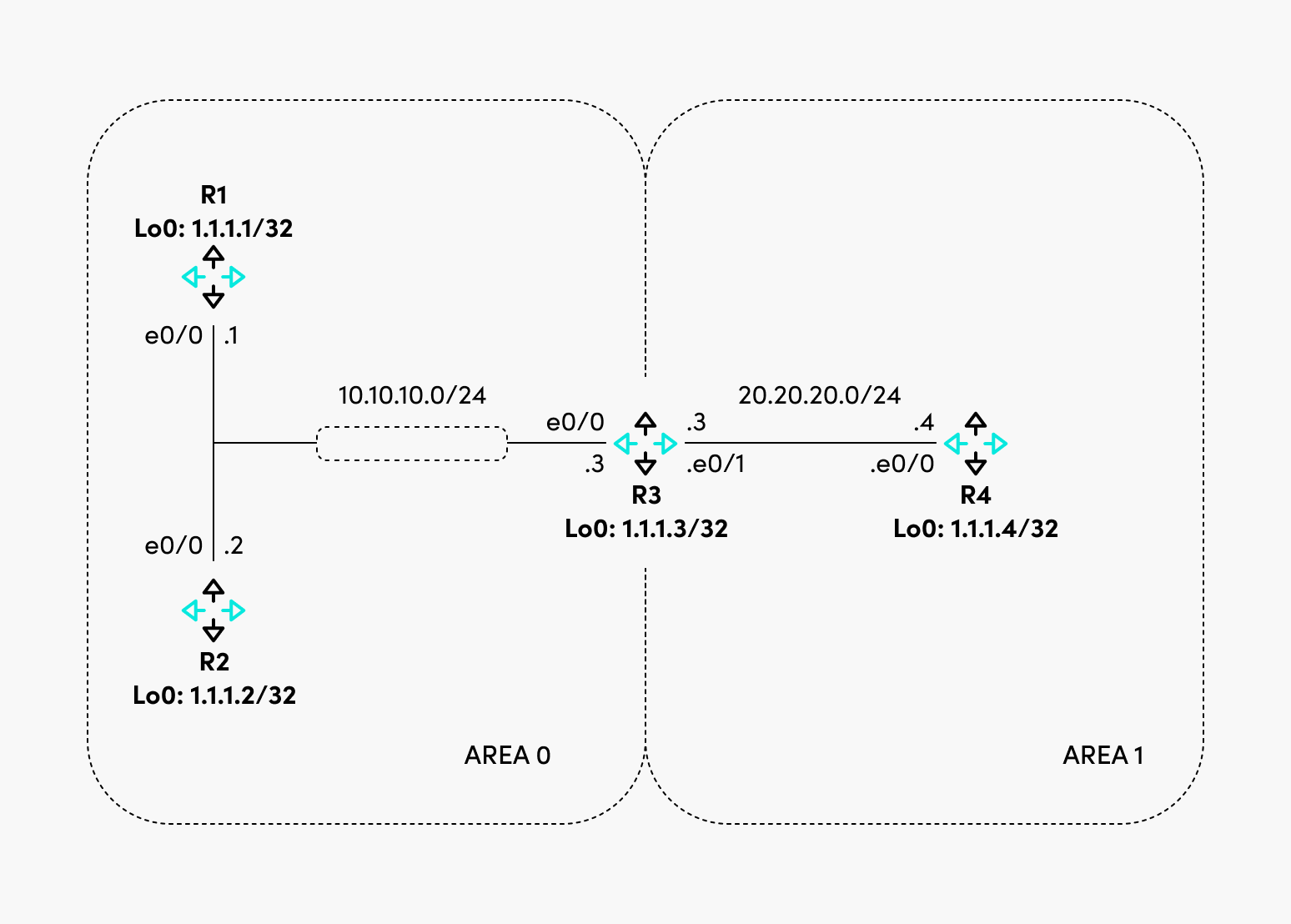


Figure 6: Multi-area OSPF.

Following interface configuration, this is what is required on R1 to be configured as an internal backbone router (the same configuration is done for R2 with the difference of IP addressing).

router ospf 1

network 1.1.1.1 0.0.0.0 area 0

network 10.10.10.1 0.0.0.0 area 0

The configuration is similar for R4, except that the interfaces are added in Area 1.

R3 has a different configuration. Observe how interfaces are part of different areas.

router ospf 1

network 1.1.1.3 0.0.0.0 area 0

network 10.10.10.3 0.0.0.0 area 0

network 20.20.20.3 0.0.0.0 area 1

Because all three routers in Area 0 were configured simultaneously and none of the OSPF parameters were changed, R3 became the DR, R2 became the BDR, and R1 became DROther. As mentioned before, the router ID of R3 was the highest because it has the highest loopback IP address out of the three routers in Area 0.

R1#show ip ospf neighbor Ethernet0/0

Neighbor ID Pri State Dead Time Address Interface

1.1.1.2 1 FULL/BDR 00:00:32 10.10.10.2 Ethernet0/0

1.1.1.3 1 FULL/DR 00:00:39 10.10.10.3 Ethernet0/0

R1#

If you checked on R3, R1 would be in the state of DROther.

R3#show ip ospf neighbor Ethernet0/0

Neighbor ID Pri State Dead Time Address Interface

1.1.1.1 1 FULL/DROTHER 00:00:35 10.10.10.1 Ethernet0/0

1.1.1.2 1 FULL/BDR 00:00:31 10.10.10.2 Ethernet0/0

R3#

Checking the routing table of R1, you would see routes from the same area (Area 0) and Area 1 (with the code of IA).

1.0.0.0/32 is subnetted, 4 subnets

O 1.1.1.2 [110/11] via 10.10.10.2, 00:06:55, Ethernet0/0

O 1.1.1.3 [110/11] via 10.10.10.3, 00:06:55, Ethernet0/0

O IA 1.1.1.4 [110/21] via 10.10.10.3, 00:06:55, Ethernet0/0

20.0.0.0/24 is subnetted, 1 subnets

O IA 20.20.20.0 [110/20] via 10.10.10.3, 00:06:55, Ethernet0/0

This would be a basic configuration of a multi-area OSPF network. This kind of deployment will suffice for most networks.

### BGP

For BGP, we will use the setup in Figure 7.

‍

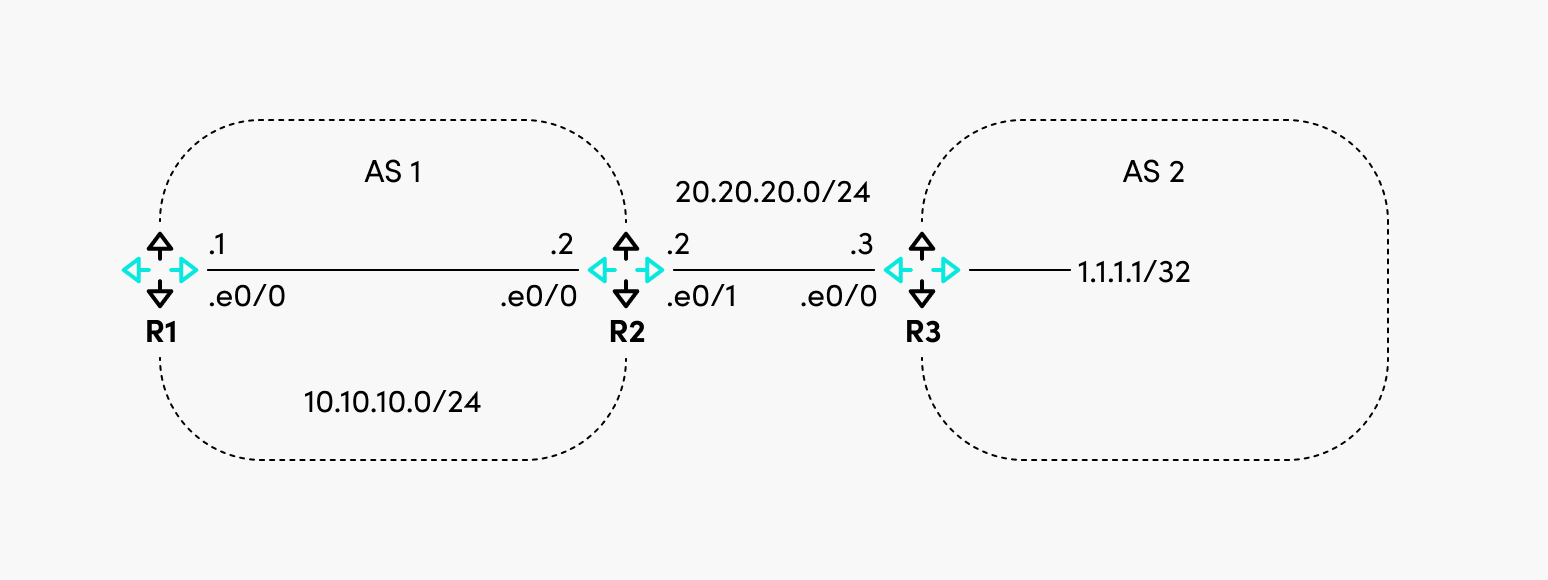


Figure 7: Internal and External BGP.

R1 and R2 are in AS1 (which means they will have an internal BGP between them), and R3 is in AS 2, so there will be an external BGP session between R2 and R3. R3 advertises the route for 1.1.1.1/32.

The configuration required for R1 is the one below.

router bgp 1

neighbor 10.10.10.2 remote-as 1

The configuration for R2 is the one below.

router bgp 1

neighbor 10.10.10.1 remote-as 1

neighbor 20.20.20.3 remote-as 2

And the configuration for R3. There is an additional command required to advertise the network 1.1.1.1/32, as noted below.

router bgp 2

network 1.1.1.1 mask 255.255.255.255

neighbor 20.20.20.2 remote-as 1

At this point, R2 should have in its routing table the network 1.1.1.1/32, but R1 will not have it because the next hop of the route (20.20.20.3) is not reachable by R1.

R1#show ip route bgp

Gateway of last resort is not set

R1#sh ip bgp 1.1.1.1

BGP routing table entry for 1.1.1.1/32, version 0

Paths: (1 available, no best path)

Not advertised to any peer

Refresh Epoch 1

2

20.20.20.3 (inaccessible) from 10.10.10.2 (10.10.10.2)

Origin IGP, metric 0, localpref 100, valid, internal

rx pathid: 0, tx pathid: 0

R1#

To solve this, we can configure R2 to set itself as the next-hop for the routes it advertises to R1.

router bgp 1

neighbor 10.10.10.1 remote-as 1

neighbor 10.10.10.1 next-hop-self

neighbor 20.20.20.3 remote-as 2

This will allow R1 to install the route in the routing table.

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

B 1.1.1.1 [200/0] via 10.10.10.2, 00:00:25

R1#

At first sight, the BGP configuration is simple, but this is as basic as it can get. In real life, one would need to play around with BGP attribute manipulation and configure additional features. Some of the attributes that can be manipulated to affect BGP routing behavior include Local Preference, AS\_PATH length, and MED to name a few. [The BGP configuration can get very complex](https://www.researchgate.net/profile/Steve-Uhlig/publication/265073133_An_Evaluation_of_BGP-based_Traffic_Engineering_Techniques/links/545892640cf2cf5164824b28/An-Evaluation-of-BGP-based-Traffic-Engineering-Techniques.pdf), and vendors constantly add new complex features to support more and more use cases.

Free Download

#### Get your copy of The SRE Report 2024

Learn the challenges facing SREs & the current state of reliability.

[**Get the Report**](https://www.catchpoint.com/asset/2024-sre-report)

## Conclusion

The following are lists that summarize the pros and cons of each routing protocol. Note that some of the cons may not actually be considered cons in certain situations, so these should be taken only as guidelines.

OSPF

* Prossome text
  + Fast convergence
  + Open standard
  + Scalable due to hierarchical area implementation
* Conssome text
  + Uses large amounts of system resources (CPU, memory) to run algorithm and maintain full network topology information
  + Multiple area types and link state advertisement types can become extensively complex

BGP

* Prossome text
  + Extremely scalable
  + Low resource usage even for large BGP tables
  + Extremely granular routing behavior adjustments
* Conssome text
  + Slow to converge
  + Complexity can increase if many BGP attributes are tweaked

BGP and OSPF are complex protocols. Their configuration can get very difficult sometimes, and it is critical to understand how the protocols work and what their core components are. Failing to do so would not only prevent you from having the right configuration, but also put you at risk of not being able to properly troubleshoot any issues with these protocols.

It is also worth checking the latest vendor documentation regarding features and their configuration, because the defaults of each feature might change from release to release.

## Chapters

**[Dynamic Routing Protocols](https://www.catchpoint.com/dynamic-routing-protocols/introduction)**

[Learn about dynamic routing protocols including the use cases, the advantages, and the limitations of RIPv2, EIGRP, OSPF, IS-IS, and BGP.](https://www.catchpoint.com/dynamic-routing-protocols/introduction)

**[1](https://www.catchpoint.com/dynamic-routing-protocols/bgp-vs-ospf)**

**[.](https://www.catchpoint.com/dynamic-routing-protocols/bgp-vs-ospf)**

**[BGP vs OSPF](https://www.catchpoint.com/dynamic-routing-protocols/bgp-vs-ospf)**

[Learn how Open Shortest Path First (OSPF) is different from Border Gateway Protocol (BGP) and understand each in more depth](https://www.catchpoint.com/dynamic-routing-protocols/bgp-vs-ospf)

**[2](https://www.catchpoint.com/dynamic-routing-protocols/eigrp-vs-ospf)**

**[.](https://www.catchpoint.com/dynamic-routing-protocols/eigrp-vs-ospf)**

**[EIGRP vs. OSPF](https://www.catchpoint.com/dynamic-routing-protocols/eigrp-vs-ospf)**

[Learn how EIGRP differs from OSPF by understanding each routing protocol in detail and by following configuration examples.](https://www.catchpoint.com/dynamic-routing-protocols/eigrp-vs-ospf)

**[3](https://www.catchpoint.com/dynamic-routing-protocols/split-horizon)**

**[.](https://www.catchpoint.com/dynamic-routing-protocols/split-horizon)**

**[Split Horizon](https://www.catchpoint.com/dynamic-routing-protocols/split-horizon)**

[This article e](https://www.catchpoint.com/dynamic-routing-protocols/split-horizon)

Routing protocols like **Open Shortest Path First (OSPF)** and **Border Gateway Protocol (BGP)** are crucial in large-scale networks, particularly for determining the most efficient paths for data transmission across complex and distributed network infrastructures. Both OSPF and BGP operate at different layers and have distinct roles in routing traffic. Here’s an in-depth explanation of how these protocols function in large-scale networks:

### ****Open Shortest Path First (OSPF)****

**OSPF** is a **link-state routing protocol** used within an **Autonomous System (AS)**, typically within a single organization or network. It is designed to efficiently handle routing decisions in large, hierarchical network topologies, making it suitable for enterprise networks.

#### **How OSPF Works:**

1. **Link-State Information Exchange**:
   * OSPF routers exchange information about the state of their links (e.g., interfaces and their statuses) to build a complete view of the network topology. Each router in the OSPF network maintains a **link-state database** (LSDB), which contains information about all the routers and their links within the network.
2. **Cost Metric**:
   * OSPF assigns a **cost** to each link based on its bandwidth. The lower the cost, the more favorable the link is. The cost is used to determine the most efficient route for packet forwarding. The default OSPF cost metric is inversely proportional to the link bandwidth (i.e., higher bandwidth links have lower costs).
3. **Shortest Path First (SPF) Algorithm**:
   * Using the link-state information, OSPF employs the **Dijkstra SPF algorithm** to calculate the shortest path from the router to each destination in the network. The result is a **routing table** that defines the best path to all reachable destinations.
4. **Hierarchical Network Design (Areas)**:
   * OSPF networks are organized into **areas**, with each area having a separate LSDB. The use of areas allows OSPF to scale in large networks by limiting the amount of routing information exchanged. There are two key types of areas:
     + **Backbone area (Area 0)**: The core of the OSPF network, where all other areas must connect.
     + **Non-backbone areas**: Subnetworks that are connected to the backbone, and only exchange routing information with it.
5. **Router Types**:
   * OSPF defines different types of routers:
     + **Internal routers**: Routers within the same OSPF area.
     + **Area Border Routers (ABR)**: Routers that connect two or more OSPF areas.
     + **Backbone Routers**: Routers within the backbone area (Area 0).
     + **Autonomous System Boundary Routers (ASBR)**: Routers that connect OSPF to external networks (e.g., BGP).
6. **Route Updates**:
   * OSPF uses **LSA (Link-State Advertisements)** to share routing information with other routers. When there is a change in the network topology (such as a router failure or a new link), OSPF routers quickly propagate these changes to other routers within the network, which then recalculate their SPF tree.
7. **Fast Convergence**:
   * OSPF is designed to quickly adapt to network changes. It uses the SPF algorithm to recalculate routes when network topology changes occur. OSPF routers can detect topology changes and recalculate paths without waiting for periodic updates, which helps in minimizing downtime.

#### **Why OSPF is Suitable for Large-Scale Networks**:

* **Scalability**: By dividing the network into multiple areas, OSPF ensures that routing information is not overloaded. This hierarchical design minimizes the amount of routing information exchanged, making it efficient for large networks.
* **Fast Convergence**: OSPF’s ability to quickly adapt to changes and recompute paths ensures minimal downtime and better network stability.
* **Flexibility**: OSPF can be used in a wide variety of network topologies and supports both IPv4 and IPv6.

### ****Border Gateway Protocol (BGP)****

**BGP** is a **path-vector protocol** that is primarily used to route traffic between different **Autonomous Systems (ASes)**, such as between different internet service providers (ISPs) or between large enterprise networks and the internet. BGP is used in **inter-domain routing**, meaning it determines how data is routed across different networks rather than within a single network.

#### **How BGP Works:**

1. **Autonomous Systems and AS Numbers**:
   * Each network that participates in BGP is assigned a unique **Autonomous System Number (ASN)**. BGP is used to exchange routing information between these ASes. Each AS is essentially an administrative domain, such as an ISP or a large organization, that manages its own routing policies.
2. **BGP Peering**:
   * BGP routers form **peer relationships** with other BGP routers, either within the same AS (iBGP - internal BGP) or across different ASes (eBGP - external BGP). Once a BGP session is established, routers exchange routing information through **BGP UPDATE messages**.
3. **Path Vector Mechanism**:
   * BGP uses a **path vector** to maintain a list of all ASes a route has passed through. This helps BGP routers make decisions based on **policy** and **reachability**. The path attribute ensures that loops do not occur, and each AS is aware of the route's history.
4. **BGP Attributes**: BGP routes are advertised with a variety of **attributes** that influence how the best path is selected. Key attributes include:
   * **AS Path**: A list of ASes the route has traversed, used for loop prevention.
   * **Next Hop**: The IP address of the next hop router that should be used to reach the destination.
   * **Local Preference**: Used within an AS to prefer certain routes over others.
   * **MED (Multi-Exit Discriminator)**: Used to influence the incoming traffic to an AS, typically in multi-homed environments.
   * **Communities**: Tags that can be attached to routes to define routing policies or preferences.
   * **Origin**: Indicates where the route originated from (e.g., IGP, EGP, or unknown).
5. **Route Selection**:
   * BGP routers select the best path based on several factors, with **AS Path length** (shorter AS Path is preferred) and **Local Preference** being some of the most important metrics. BGP also supports load balancing, so multiple paths can be used if the network allows it.
6. **Policy-Based Routing**:
   * Unlike OSPF, which focuses on the shortest path, BGP allows network administrators to influence route selection based on policy. This includes preferring certain ISPs, controlling traffic flows based on business needs, or balancing load across multiple links.
7. **Convergence**:
   * BGP has slower convergence compared to IGPs like OSPF, meaning it may take longer to detect and adapt to network changes. However, BGP is more stable over the long term, as it allows for the manual control of routing policies, making it ideal for large-scale networks with complex routing needs.

#### **Why BGP is Suitable for Large-Scale Networks**:

* **Scalability**: BGP is designed to handle routing between a vast number of ASes, making it the foundation of the internet. It supports a large number of routes and is scalable to handle the global routing table.
* **Policy Control**: BGP allows fine-grained control over routing decisions, enabling administrators to implement traffic engineering and optimize the flow of data across multiple ASes.
* **Loop Prevention and Redundancy**: BGP uses mechanisms like the AS Path to avoid routing loops and supports redundancy by advertising multiple routes for the same destination.

### ****Key Differences Between OSPF and BGP in Large-Scale Networks****

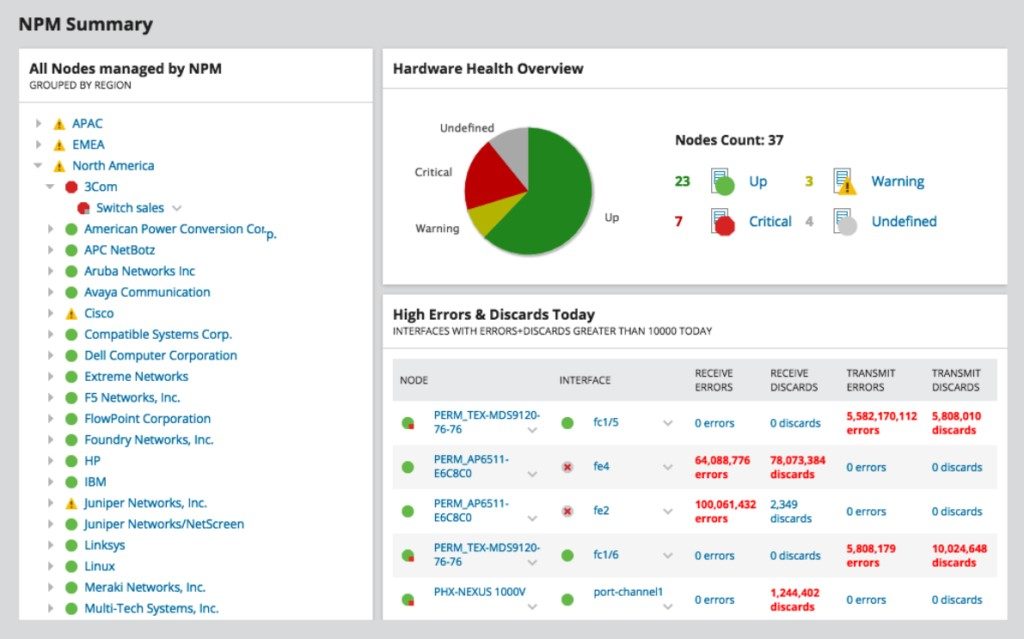
| **Feature** | **OSPF** | **BGP** |
| --- | --- | --- |
| **Type of Routing** | Interior Gateway Protocol (IGP) – used within a single AS | Exterior Gateway Protocol (EGP) – used between different ASes |
| **Routing Metric** | Cost (based on bandwidth) | Path attributes (e.g., AS Path, Local Preference, MED) |
| **Scalability** | Good for large networks within a single AS | Extremely scalable for routing between multiple ASes (Internet-scale) |
| **Convergence Time** | Fast convergence, recalculates quickly on topology changes | Slower convergence, especially when topology changes occur |
| **Routing Policy** | Relatively simple, focuses on shortest path | Highly flexible, allows complex routing policies based on attributes |
| **Hierarchical Design** | Supports hierarchical area-based design | No hierarchical design, relies on path attributes for routing decisions |
| **Network Type** | Primarily used in large enterprise or service provider networks | Used for inter-AS routing, often for connecting different ISPs and large enterprises to the internet |

### ****Conclusion****

In large-scale networks, **OSPF** and **BGP** serve complementary roles. **OSPF** is ideal for routing within a single AS, offering

### ****Conclusion****

In large-scale networks, **OSPF** and **BGP** serve complementary roles. **OSPF** is ideal for routing within a single AS, offering fast convergence and efficient path selection in an intra-domain environment. It’s especially useful in large enterprise networks where there are multiple routers and a need for quick, dynamic adjustments. On the other hand, **BGP** is crucial for routing between different ASes, such as between ISPs or between an organization and the internet, and it supports the vast, policy-driven routing that is characteristic of the internet backbone. Both protocols are essential for maintaining performance, scalability, and reliability in large-scale network infrastructures.



©2023 SolarWinds Worldwide, LLC. All rights reserved.

Other NPM features include [NetPath](https://www.solarwinds.com/network-performance-monitor/use-cases/netpath?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-NPM-20190923_NetworkTroubles_X_X_VidNo_X-X" \t "_blank)[™](https://www.solarwinds.com/network-performance-monitor/use-cases/netpath?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-NPM-20190923_NetworkTroubles_X_X_VidNo_X-X" \t "_blank)[network path analysis](https://www.solarwinds.com/network-performance-monitor/use-cases/netpath?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-NPM-20190923_NetworkTroubles_X_X_VidNo_X-X" \t "_blank), which lets you see your network topology and performance pathways, and [PerfStack](https://www.solarwinds.com/perfstack?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-PFSTK-20190923_NetworkTroubles_X_X_VidNo_X-X" \t "_blank)™, which allows you to compare different performance metrics against each other, as well as historical data. With these tools, you can see which performance issues may be interlinked and troubleshoot the root cause faster. NPM also comes with tools like [Wi-Fi sniffer](https://www.solarwinds.com/network-performance-monitor/use-cases/wifi-packet-sniffer?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-NPM-20190923_NetworkTroubles_X_X_VidNo_X-X), software for monitoring load balancers, switches, and firewalls, as well as wireless issues and coverage, all of which enables you to keep an eye on the overall health of your network and quickly pinpoint and fix issues as soon as they arise.

## Troubleshooting Network Issues Conclusions

Network troubleshooting can be stressful at the best of times. Having clear steps to follow, an arsenal of best practices and a robust monitoring tool like [Network Performance Monitor](https://www.solarwinds.com/network-performance-monitor?CMP=ORG-BLG-DNS-X_WW_X_NP_X_X_EN_X_X-NPM-20190923_NetworkTroubles_X_X_VidNo_X-X) can help make the process as smooth as possible.

In Linux, package management tools are essential for installing, updating, removing, and managing software packages on a system. Each Linux distribution has its own package management tools, depending on the package format and the underlying system architecture. These tools ensure that software is installed correctly and that dependencies are managed automatically. Below is an overview of different Linux package management tools, their associated package formats, and when each would be used.

**1. APT (Advanced Package Tool) - Used in Debian-based distributions (e.g., Ubuntu, Debian)**

**Package Format**: .deb

**Key Tools**:

* apt-get: The traditional tool for managing packages in Debian-based distributions.
* apt: A newer, more user-friendly tool for managing packages.
* dpkg: A low-level tool for handling .deb packages directly.

**Usage**:

* **Installing Software**: apt install package\_name
* **Removing Software**: apt remove package\_name
* **Updating Software**: apt update (to update the package list) and apt upgrade (to upgrade installed packages).
* **Searching for Packages**: apt search package\_name
* **Dependency Management**: APT automatically resolves and installs dependencies when installing software.

**When to Use**:

* APT is used in **Debian-based distributions** such as Ubuntu, Linux Mint, and others.
* It's most useful for users of these distributions because it is integrated into the system for easy installation and management of software packages, security updates, and repositories.

**2. YUM (Yellowdog Updater, Modified) - Used in Red Hat-based distributions (e.g., CentOS, Fedora, RHEL)**

**Package Format**: .rpm

**Key Tools**:

* yum: The default package manager for older versions of Red Hat-based distributions (up to RHEL/CentOS 7).
* dnf: The next-generation package manager, which replaced yum in RHEL/CentOS 8 and Fedora.

**Usage**:

* **Installing Software**: yum install package\_name or dnf install package\_name
* **Removing Software**: yum remove package\_name or dnf remove package\_name
* **Updating Software**: yum update or dnf update
* **Searching for Packages**: yum search package\_name or dnf search package\_name
* **Viewing Installed Packages**: yum list installed or dnf list installed

**When to Use**:

* **YUM** is used in **Red Hat-based distributions**, such as CentOS, RHEL (Red Hat Enterprise Linux), and older versions of Fedora.
* **DNF** is the replacement for YUM in RHEL/CentOS 8, Fedora, and newer versions.
* These tools are appropriate for managing software on enterprise and server systems running Red Hat, CentOS, or Fedora.

**3. Zypper - Used in openSUSE and SUSE Linux Enterprise**

**Package Format**: .rpm

**Key Tools**:

* zypper: The primary package management tool used in openSUSE and SUSE Linux distributions.

**Usage**:

* **Installing Software**: zypper install package\_name
* **Removing Software**: zypper remove package\_name
* **Updating Software**: zypper update
* **Searching for Packages**: zypper search package\_name
* **Managing Repositories**: zypper addrepo and zypper removerepo

**When to Use**:

* **Zypper** is used in **openSUSE** and **SUSE Linux Enterprise**.
* It's appropriate for managing software packages, updates, and repositories on systems running these distributions, especially in enterprise environments.

**4. Pacman - Used in Arch Linux and Arch-based distributions (e.g., Manjaro)**

**Package Format**: .pkg.tar.xz (or .pkg.tar.zst in newer versions)

**Key Tools**:

* pacman: The package manager for Arch Linux and its derivatives.

**Usage**:

* **Installing Software**: pacman -S package\_name
* **Removing Software**: pacman -R package\_name
* **Updating Software**: pacman -Syu (update the system)
* **Searching for Packages**: pacman -Ss package\_name
* **Viewing Installed Packages**: pacman -Q

**When to Use**:

* **Pacman** is used in **Arch Linux** and **Arch-based distributions** such as Manjaro, ArcoLinux, and EndeavourOS.
* It’s the go-to tool for users of Arch Linux who prefer a rolling release distribution and a more hands-on approach to system management.

**5. Snap - Universal Package Manager**

**Package Format**: .snap

**Key Tools**:

* snap: A universal package management system that works across many Linux distributions.

**Usage**:

* **Installing Software**: snap install package\_name
* **Removing Software**: snap remove package\_name
* **Updating Software**: snap refresh
* **Searching for Packages**: snap find package\_name
* **Viewing Installed Snap Packages**: snap list

**When to Use**:

* **Snap** is suitable when you need a **cross-distribution package** that works on various Linux distributions (e.g., Ubuntu, Fedora, Debian, CentOS).
* It is used for installing software that is bundled with all dependencies inside a single package, making it ideal for distributing software across different Linux systems.

**6. Flatpak - Universal Package Manager**

**Package Format**: .flatpak

**Key Tools**:

* flatpak: A universal package manager that is cross-distribution like Snap.

**Usage**:

* **Installing Software**: flatpak install flathub package\_name
* **Removing Software**: flatpak uninstall package\_name
* **Updating Software**: flatpak update
* **Searching for Packages**: flatpak search package\_name
* **Viewing Installed Flatpak Packages**: flatpak list

**When to Use**:

* **Flatpak** is another universal packaging system designed for Linux distributions. It allows you to install applications in isolation, with all necessary dependencies bundled within the package.
* It's often used in **GNOME**-based systems, and distributions like Fedora, Ubuntu, and others support it.
* It’s suitable for software that requires a self-contained runtime and cross-distribution support.

**7. RPM (Red Hat Package Manager) - Low-level Tool for RPM-Based Systems**

**Package Format**: .rpm

**Key Tools**:

* rpm: The low-level package management tool for .rpm packages.

**Usage**:

* **Installing Software**: rpm -i package\_name.rpm
* **Removing Software**: rpm -e package\_name
* **Querying Installed Packages**: rpm -q package\_name
* **Verifying Package Integrity**: rpm -V package\_name

**When to Use**:

* **RPM** is used directly for handling .rpm packages and is often invoked when you need more control over the installation or removal of a package, such as in custom or local package installations.
* It is not typically used for daily software management, as higher-level tools like YUM, DNF, or Zypper are preferred for convenience and handling dependencies automatically.

**8. AppImage - Portable Software Packaging Format**

**Package Format**: .AppImage

**Key Tools**:

* **AppImage**: A format for distributing software in a single, portable file.

**Usage**:

* **Running Software**: Make the file executable (chmod +x package\_name.AppImage) and run it directly.
* **No installation required**: AppImages do not need to be installed; they run directly from their location on the system.

**When to Use**:

* **AppImage** is used when you need a **self-contained application** that runs across various Linux distributions without installation.
* It's great for software developers who want to provide their applications in a format that doesn’t depend on a specific package manager.

**Conclusion:**

* **APT**, **YUM**, **Zypper**, **Pacman**, and **RPM** are the traditional package management tools for different Linux distributions, each suited to a particular ecosystem (Debian-based, Red Hat-based, Arch-based, etc.).
* **Snap** and **Flatpak** are cross-distribution tools that bundle all dependencies with the application, making them ideal for developers and users who need software to run consistently across different distributions.
* **AppImage** is a unique portable format that requires no installation and is useful for running software in a self-contained manner.

The choice of package management tool depends on the Linux distribution you’re using, your need for portability, and the type of software you're installing. For most users, the system's default package manager (like apt, yum, or zypper) is the most appropriate tool. However, for those who want to use software across multiple distributions or prefer a single, all-inclusive package, **Snap**, **Flatpak**, and **AppImage** provide universal solutions.

# An Overview of Package Management in Linux

Published September 5, 2023 by [Linode](https://www.linode.com/docs/contributors/linode/)

[**Create a Linode account**](https://login.linode.com/signup?promo=docs07082024) to try this guide with a $100 credit.

*This credit will be applied to any valid services used during your first 60 days.*

Top of Form

Sign Up

Bottom of Form

On Linux, software is typically built as a package, distributed through repositories, and managed on the end-user’s system through package managers. Each Linux system typically contains thousands of packages, many of which are required dependencies for other packages.

Many guides within Linode’s documentation (and elsewhere online) require the installation of new software. These guides typically provide basic commands that utilize a package manager to install the package(s) corresponding to that software. In some cases, you may wish to go beyond these basic commands to install a particular version, search for previously installed packages, or perform other actions. The purpose of this guide is to provide a solid understanding of package management in Linux and an overview of the most widely used package managers.

## Core Concepts for Package Management

### Packages

Most software applications designed for Linux or Unix systems are distributed as packages, which are archives that contain the pre-compiled binary software files, installation scripts, configuration files, dependency requirements, and other details about the software. These packages are typically specific to a particular distribution and formatted in that distribution’s preferred package format, such as .deb for Debian/Ubuntu and .rpm for CentOS/RHEL.

While it’s relatively simple for a user to install a package file, there are other complexities to consider. These complexities include obtaining (downloading) the package, ensuring packages are upgraded with security and bug fixes, and maintaining all the dependencies for the software.

### Repositories

While a user can obtain package files through any method of file transfer, it is typical to use software repositories (also called repos) to obtain packages. Repositories are simply the location where the packages are stored, commonly accessible via the internet. A repository can contain a single package or thousands of packages. Most Linux distributions have their own unique repositories, sometimes separating out their cores packages into one while additional features are in others.

### Dependencies

Its very common in almost any operating system for software to require other software to run. In Linux, each package contains metadata detailing the additional packages that are required. These additional packages are called dependencies. A single package can sometimes have hundreds of dependencies. When installing, upgrading, or removing packages, these dependencies may also need to installed, upgraded, and optionally removed.

### Package Managers

A package manager reduces the complexity for the end-user by automating the process of obtaining, installing, upgrading, and removing packages and their dependencies. This dramatically improves the user experience and the ability to properly and efficiently manage the software on your Linux system. Today, package managers can be a defining feature for Linux distributions and many system administrators prefer to use a particular distribution based on its package management system (among other considerations).

## Benefits of Package Management Systems

* **Easily obtain the correct, trusted, and stable package for your Linux distribution.** Since most distributions maintain their own repositories, using a package manager can ensure you only install packages that have been thoroughly vetted, are stable, are trusted, and work with your system. The subjective judgments and community standards that guide package management also guide the “feel” and “stability” of a given system.
* **Automatically manage all dependencies when taking action on a package.** While dependencies can be distributed inside the original package to ensure the correct versions of the required software are always present, this increases disk usage and package file size. In most cases, package managers will manage dependencies as separate packages, allowing them to be shared with other software and reducing disk usage and file size.
* **Follow the unique conventions for each Linux distribution.** Linux distributions often have conventions for how applications are configured and stored in the /etc/ and /etc/init.d/ directories. By using packages, distributions are able to enforce a single standard.
* **Stay up-to-date on security patches and software updates.** Most package managers provide a single command to automatically update all packages to the latest versions stored on the configured repositories. Provided those repositories are consistently maintained, this enables you to quickly get important security updates.

## Comparison of Package Managers

There are lots of package managers in Linux, each working a bit differently. Here is a list of common package managers, along with their supported distributions, package file formats, and a description.

### APT

[Using APT to Manage Packages in Debian and Ubuntu](https://www.linode.com/docs/guides/apt-package-manager/)

* **Distributions:** Ubuntu, Debian, and Kali Linux
* **Commands:** apt, apt-get, apt-cache
* **Underlying package management tool:** [dpkg](https://linux.die.net/man/1/dpkg" \t "_blank)
* **Package file format:** .deb

Advanced Package Tool, more commonly known as [APT](https://ubuntu.com/server/docs/package-management), is a package management system for Debian, Ubuntu, and other similar Linux distributions. It acts as a front-end to the lower-level **[dpkg](https://en.wikipedia.org/wiki/Dpkg" \t "_blank)** package manager, which is used for installing, managing, and providing information on .deb packages. Most distributions that use APT also include a collection of command-line tools that can be used to interface with APT. These tools include apt-get, apt-cache, and the newer apt, which essentially combines both of the previous tools with some modified functionality.

### DNF

[Using DNF to Manage Packages in CentOS/RHEL 8 and Fedora](https://www.linode.com/docs/guides/dnf-package-manager/)

* **Distributions:** RHEL/CentOS 8, Fedora 22, and later versions of both distributions
* **Commands:** dnf, yum
* **Underlying package management tool:** [RPM](https://linux.die.net/man/8/rpm) (RPM Package Manager)
* **Package file format:** .rpm

Dandified YUM, or simply [DNF](https://fedoraproject.org/wiki/DNF?rd=Dnf), is the successor to YUM. Just like YUM, DNF provides a user-friendly interface to the RPM Package Manager (RPM) that comes with CentOS, RHEL, Fedora, and many other Linux distributions. As the successor to YUM, DNF has several enhancements including increased performance, faster dependency resolution, and more complete documentation for its API. Most distributions still link the yum command to the DNF software and, since DNF maintains compatibility with much of YUM’s CLI, most commands using yum still function as intended.

### YUM

[Using YUM to Manage Packages in CentOS/RHEL 7 and Earlier](https://www.linode.com/docs/guides/yum-package-manager/)

* **Distributions:** RHEL/CentOS 7, Fedora 21, and earlier versions of both distributions
* **Command:** yum
* **Underlying package management tool:** [RPM](https://linux.die.net/man/8/rpm) (RPM Package Manager)
* **Package file format:** .rpm

Yellowdog Updater, Modified, more commonly known as [YUM](http://yum.baseurl.org/), is a package management tool for a variety of older RHEL-based distributions (such as CentOS 7) and older versions of Fedora. It provides an easy-to-use interface on top of the low-level functions available in the RPM Package Manager (RPM). It has largely been replaced by it successor Dandified YUM, also called DNF, on most newer RPM-based distributions.

### Zypper

[Use Zypper to Manage Packages in openSUSE](https://www.linode.com/docs/guides/zypper-package-manager/)

* **Distributions:** openSUSE
* **Command:** zypper
* **Underlying package management tool:** ZYpp (also called [libzypp](https://doc.opensuse.org/projects/libzypp/HEAD/" \t "_blank))
* **Package file format:** .rpm

[Zypper](https://en.opensuse.org/Portal:Zypper) is the CLI tool used for managing packages on openSUSE Linux distributions. Just like DNF and YUM, packages are stored in the .rpm file format, though Zypper interfaces with the ZYpp (libzypp) library instead of RPM. Some users report that Zypper is faster than other package managers and, unlike many others, has the ability to add repositories directly from its own CLI. See the [Zypper manual](https://en.opensuse.org/SDB:Zypper_manual" \t "_blank) for more usage details.

### Pacman

[Using Pacman to Manage Packages in Arch](https://www.linode.com/docs/guides/pacman-package-manager/)

* **Distributions:** Arch-based, including Arch and Manjaro
* **Command:** pacman
* **Package file format:** .tar.xz (and other compressed tar formats)

Arch Linux and other similar distributions (like the popular Manjaro desktop distro) use the [pacman](https://wiki.archlinux.org/title/pacman" \t "_blank) package manager. Packages are stored as compressed tarballs and, as such, are generally smaller than other package formats. Pacman is unique in that it comes with a system to build packages, not just manage them. This system is called the ABS (Arch Build System).

### Portage

[Using Portage to Manage Packages in Gentoo](https://www.linode.com/docs/guides/portage-package-manager/)

* **Distributions:** Gentoo
* **Command:** emerge
* **Package file format:** ebuild shell script or .tbz2 (compressed tar archive)

[Portage](https://wiki.gentoo.org/wiki/Portage), the package manager for Gentoo, is quite a bit different than other solutions. Instead of using pre-compiled binary packages, Portage packages are typically shell scripts called **ebuilds**. The [emerge](https://dev.gentoo.org/~zmedico/portage/doc/man/emerge.1.html) CLI tool can run these shell scripts to install packages, and is also responsible for managing dependencies and a database of installed packages.

### Slackware Package Management

[Managing Packages in Slackware](https://www.linode.com/docs/guides/slackware-package-management/)

* **Distributions:** Slackware
* **Commands:** slackpkg, pkgtool, installpkg, upgradepkg, removepkg
* **Package file format:** .tgz and .txz (compressed tar archive)

Slackware comes bundled with a few package management tools. The **pkgtool** is a TUI (menu-driven text interface) for managing packages and installing local packages. To install packages located on the internet, the **slackpkg** tool can be used. For more advanced tasks, use specialized tools like **installpkg**, **upgradepkg**, and **removepkg**.

This page was originally published on September 5, 2023.

linux

The **Address Resolution Protocol (ARP)** is a network protocol used to map a **logical IP address** to a **physical MAC address** in a local area network (LAN). It operates at the **Data Link Layer** of the OSI model, bridging the gap between the network layer (where IP addresses are used) and the data link layer (where MAC addresses are used).

**Purpose of ARP:**

1. **IP to MAC Address Mapping**:
   * ARP's primary purpose is to allow devices to discover the **MAC address** associated with a given **IP address** on the same local network. Devices use MAC addresses to communicate directly with each other on the data link layer, while IP addresses are used for routing at the network layer.
2. **Facilitates Communication in a Local Network**:
   * When a device needs to send data to another device on the same network, it must know the recipient's MAC address. If the sending device only knows the recipient's IP address, it uses ARP to resolve the corresponding MAC address.
3. **ARP Request and Reply**:
   * If the sender doesn't have the MAC address in its ARP cache, it broadcasts an **ARP request** on the local network, asking "Who has this IP address?" The device with the matching IP address responds with an **ARP reply**, which includes its MAC address.
4. **Caching for Efficiency**:
   * Once the MAC address is learned, it is stored in the **ARP cache** of the sending device for future use. This reduces the need for repeated ARP requests for the same IP addresses.
5. **Enables Layer 2 Communication**:
   * ARP is crucial for enabling Layer 2 (Data Link Layer) communication, which uses MAC addresses for actual data transfer between devices on the same network segment.

**Example:**

* When Host A (IP: 192.168.1.10) needs to send data to Host B (IP: 192.168.1.20), and Host A knows Host B's IP but not its MAC address, Host A sends an ARP request. Host B replies with its MAC address, allowing Host A to encapsulate the data in a frame with the correct destination MAC address.

In summary, ARP is essential for mapping IP addresses to MAC addresses and ensuring that devices can communicate effectively within a local networ

# What’s ARP? Address Resolution Protocol Explained



By [Shanika Wickramasinghe](https://www.splunk.com/en_us/blog/author/shanika-wickramasinghe.html)

[Share on X](https://twitter.com/intent/tweet?text=Found%20this%20useful%20link%20for%20you.%20%23splunk&url=https%3A%2F%2Fwww.splunk.com%2Fen_us%2Fblog%2Flearn%2Faddress-resolution-protocol-arp.html" \o "Share on X" \t "_blank)

[Share on Facebook](https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.splunk.com%2Fen_us%2Fblog%2Flearn%2Faddress-resolution-protocol-arp.html&title=Found%20this%20useful%20link%20for%20you." \o "Share on Facebook" \t "_blank)

[Share on LinkedIn](http://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fwww.splunk.com%2Fen_us%2Fblog%2Flearn%2Faddress-resolution-protocol-arp.html&title=Share%20on%20LinkedIn&summary=Found%20this%20useful%20link%20for%20you." \o "Share on LinkedIn" \t "_blank)

When you connect to the internet, you get a logical identifier for your device called an IP address. However, for actual communication to happen within a local area network, the MAC address of a device is needed.

The Address Resolution Protocol (ARP) is a fundamental network communication protocol used within LANs to discover the MAC address of a device using its known IP address. The ARP protocol operates at the Data Link layer of [the OSI model](https://www.splunk.com/en_us/blog/learn/osi-model.html).

Various IT roles — network administrators, [cybersecurity analysts](https://www.splunk.com/en_us/blog/learn/security-analyst-role-responsibilities.html), [system administrators](https://www.splunk.com/en_us/blog/learn/system-administrator-sysadmin-role.html), IT technicians, and [penetration testers](https://www.splunk.com/en_us/blog/learn/penetration-testing.html) — need to have a good understanding of this concept.

Let’s take a look.

## The purpose of ARP

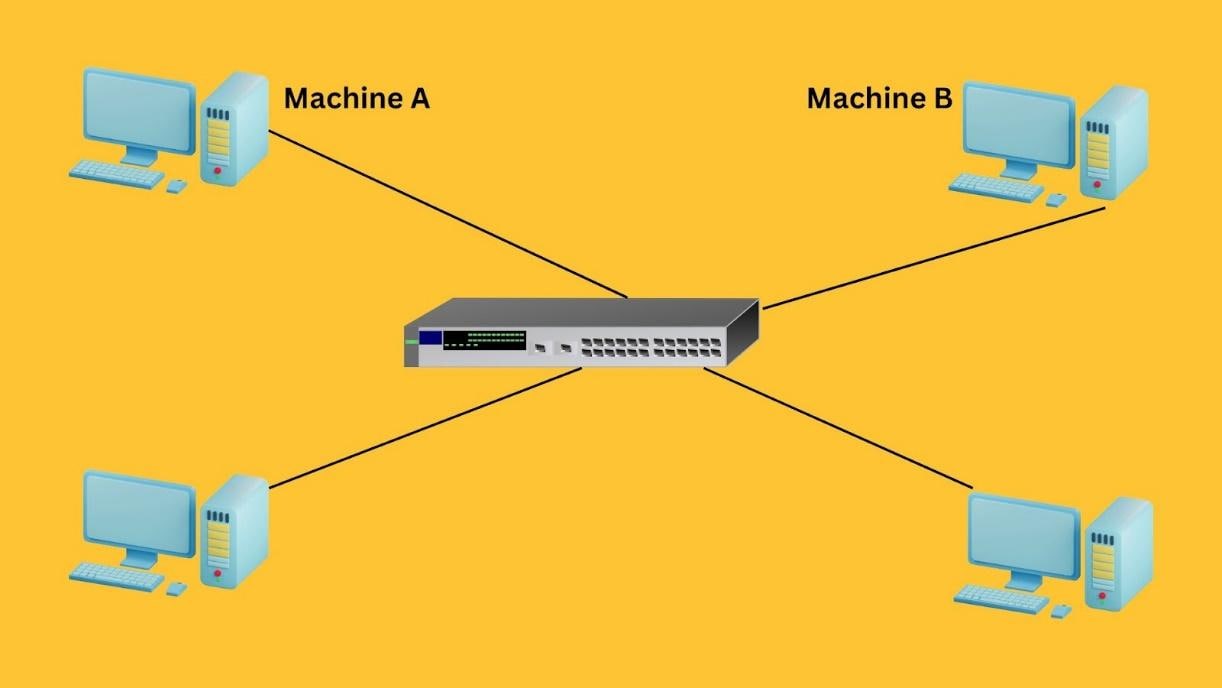
In the early 1980s, Ethernet was becoming a popular local area network (LAN) technology operating at the Data Link layer, dealing with MAC addresses. Meanwhile, [TCP/IP](https://www.splunk.com/en_us/blog/learn/tcp-ip.html) operated at the Network layer, using IP addresses.

Because of this separation, a dynamic solution was required to resolve and maintain IP-to-MAC address associations. Developed in 1982 by researchers David C. Plummer and Bill Simpson, ARP was introduced to map IP addresses to MAC addresses in local networks.

## How the ARP Protocol works

Local area networks (LANs) are a very common [network configuration](https://www.splunk.com/en_us/blog/learn/network-configuration.html). The following image shows a LAN where Machine A wants to communicate with Machine B.

Using this example, let's understand how ARP works.



[Here](https://www.whatismyip.com/arp-address-resolution-protocol/), Machine A wants to communicate with Machine B. Suppose Machine A already knows the IP address of Machine B. However, for this communication to take place, Machine A still needs Machine B’s MAC address.

Machine A will first look at its internal list — called the ARP cache — to see if Machine B’s IP address has a matching MAC address. If no entries are found, Machine A can use the known IP address of Machine B to locate the specific device on the network.

To do that, Machine A will send out a broadcast message to all the machines on the network, asking every device which machine has this particular IP address and requesting its associated MAC address.

The machine with the matching IP address will respond back and share its MAC address with Machine A. Once Machine A receives Machine B’s MAC address, communication can take place.

### Steps in an ARP request

Based on the above example, I’ve translated how ARP works into these clear steps.

**Step 1: Initial cache check.**When a sender wishes to communicate with a receiver, it first checks its ARP cache to determine if it already contains the MAC address associated with the receiver’s IP address. If the MAC address is present, communication proceeds directly using that address.

**Step 2: Preparing the ARP request.**If the MAC address is not in the cache, the sender prepares an ARP request. This packet includes the sender’s MAC and IP addresses, with the receiver's IP address specified and the MAC address field left blank as it is unknown.

**Step 3: Broadcasting the ARP request.**The sender then broadcasts this ARP request across the local area network. Every device connected to the LAN receives this broadcast.

**Step 4: ARP request evaluation by LAN devices.**Upon receiving the ARP request, all the device compares the requested IP address by the host with its own.

* Devices with non-matching IP addresses discard the request.
* The device with the matching IP address processes it further.

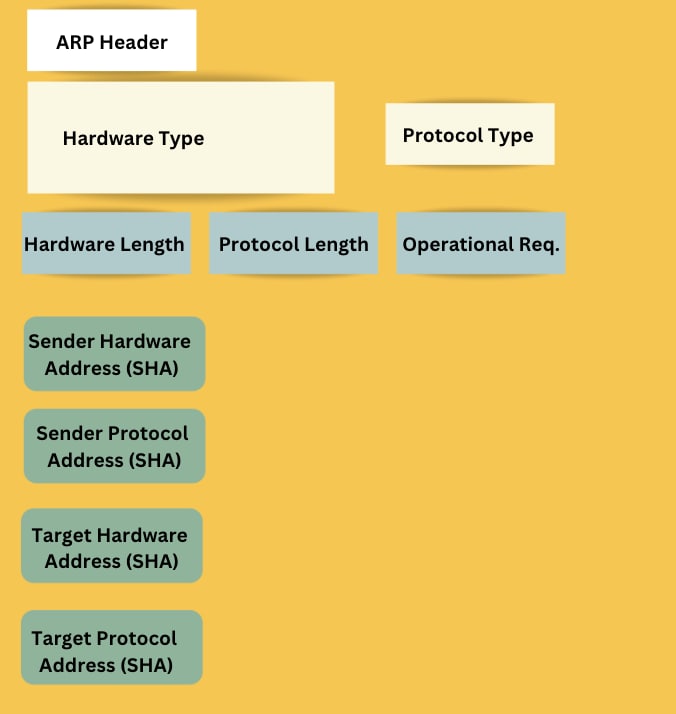
**Step 5: Responding to the ARP request.** The device that contains the matching IP address prepares an ARP reply. This unicast message, containing both its IP and MAC addresses, is sent directly back to the sender.

**Step 6: Updating the ARP cache.**Once the sender receives this ARP reply, it updates its ARP cache with the new MAC address information. This update allows the sender to communicate directly with the receiver using the resolved MAC address.

**Step 7: Communication.**With the MAC address now known and recorded in the ARP cache, the sender can send data packets directly to the receiver.

## Message format of ARP protocol

The following image shows the main components that an ARP message consists of.



* **ARP Header** specifies the message type, such as 'Request' or 'Reply', and includes address length fields.
* **Hardware Type** is a 2-byte field indicating the hardware used for transmission, typically Ethernet (value 1).
* **Protocol Type** indicates the protocol used, often IPv4 (value 2048).
* **Hardware Address Length** shows the length of the hardware address in bytes; for Ethernet, this is 6 bytes.
* **Protocol Address Length** displays the size of the protocol address, with the IP address size being 4 bytes.
* **OP Code** specifies the operation; '1' for request and '2' for reply, dictating the message’s purpose.
* **Sender hardware address** contains the sender’s MAC address, identifying the sender’s network adapter.
* **Sender protocol address** holds the sender’s IP address, associated with the sender’s MAC address.
* **Target Hardware Address.** For requests, this is zero-filled since the MAC address is unknown; for replies, it contains the target device's MAC address.
* **Target Protocol Address** contains the IP address of the device whose MAC address is being queried.

## Four types of ARP

We can recognize different types of ARP based on how they have been implemented and the use cases they have in different scenarios.

### Dynamic ARP

Dynamic ARP entries are commonly used in local area networks (LANs). These entries are automatically created and managed by the ARP protocol during normal network communication.

These entries have a limited lifespan and will expire after a certain period if not used again.

### Static ARP

Network administrators use Static ARP when they want to manually set fixed IP-to-MAC address mappings. These entries do not expire or get affected by the typical updating processes of dynamic ARP.

This stability increases security and is particularly useful in preventing ARP spoofing attacks.

### Proxy ARP

Proxy ARP enables a Layer 3 device, such as a router, to respond to ARP requests on behalf of another device that is on a different subnet. This method involves the router mapping its own MAC address to the IP address of the target device, thereby misleading the sender into believing it has reached its intended destination directly.

Network administrators commonly use proxy ARP to:

* Connect subnets.
* Handle legacy systems.
* Increase security by hiding [network topology](https://www.splunk.com/en_us/blog/learn/network-topology.html).

### Gratuitous ARP

Gratuitous ARP is a proactive broadcast by a device to update other devices on the network about its IP-to-MAC address mapping. Gratuitous ARP has two purposes:

* **Updating the ARP caches** of other devices to prevent outdated entries.
* **Checking for duplicate** IP addresses on the network.

When a device broadcasts a Gratuitous ARP request for its own IP address, if no responses are received, it confirms that no other device is using the same IP. However, if a response is received, it indicates that another device is using that IP which leads to address conflicts.

Thus, Gratuitous ARP helps maintain network integrity and prevent IP address duplication.

## Advantages & challenges of ARP

|  |  |
| --- | --- |
| **Advantages** | **Challenges** |
| Increases the efficiency of network communication by translating IP addresses to MAC addresses. | Susceptible to [security vulnerabilities](https://www.splunk.com/en_us/blog/learn/vulnerability-types.html) such as ARP spoofing. |
| Automatically updates its cache to adapt to network changes. | ARP requests can lead to [network congestion](https://www.splunk.com/en_us/blog/learn/network-traffic-congestion.html), especially in large networks. |
| ARP enables dynamic mapping between IP and MAC addresses. This simplifies network administration and allows easy addition or removal of devices without manual setup. | Lacks mechanisms to [authenticate](https://cms.splunk.com/editor.html/content/splunk-blogs/en_us/category/learn/authenticate) and verify device identities. |
| The IP-to-MAC address mappings feature of ARP helps network administrators identify and resolve connectivity issues. |  |

*(Related reading:*[*network monitoring*](https://www.splunk.com/en_us/blog/learn/network-monitoring.html)*,*[*the network operations center*](https://www.splunk.com/en_us/blog/learn/noc-network-operations-center.html)*&*[*network security*](https://www.splunk.com/en_us/blog/learn/network-security.html)*.)*

## ARP commands & usage

There [is a set](https://www.tutorialspoint.com/arp-commands) of ARP commands that can be used to deal with the ARP cache. Network administrators need to know these commands as they help diagnose and resolve network connectivity issues related to ARP.

Here are some ARP commands:

User:>arp

This command displays the cache, including the IP address, physical address, and type of each entry in the cache.

User:>arp -s 172.16.1.10 00-11-55-33-66-44

Use the **-s option** to add an entry to the cache.

User:>arp -d 172.16.1.10

Here **-d option** is used to delete an entry available in the cache.

User:>arp -a

Here **-a option** displays the machine’s ARP cache. The output lists IP addresses, MAC addresses, and the type of each entry which can be either dynamic or static.

## What is ARP spoofing?

I mentioned ARP Spoofing earlier. Common in [man in the middle attacks](https://www.splunk.com/en_us/blog/learn/man-in-the-middle-attacks.html), ARP spoofing is a technique where attackers intercept communication between network devices. In this scenario, an attacker who has access to the network will send false ARP messages. The purpose is to link their own MAC address with the IP address of a device on the network.

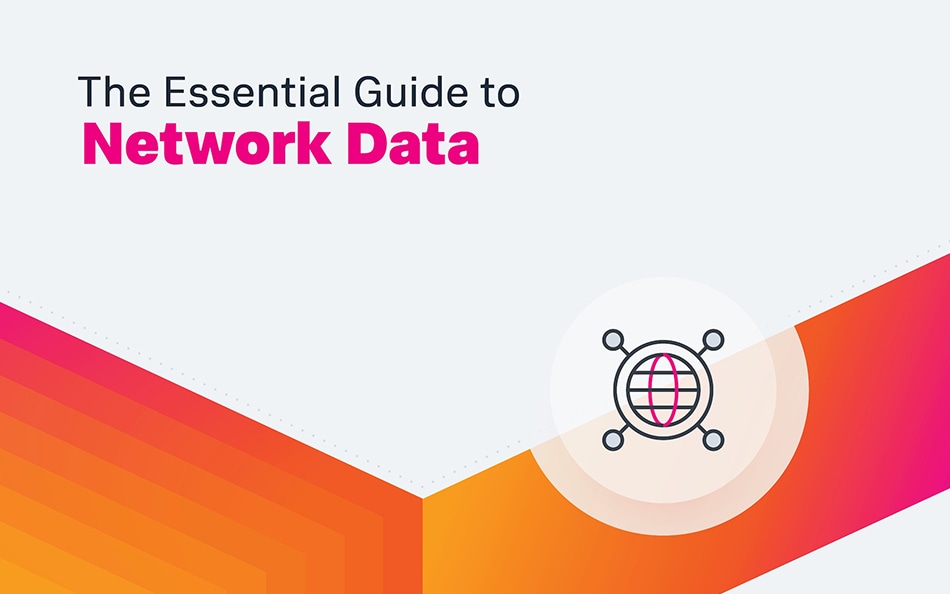
By using this technique, attackers can misdirect data meant for another device to their own device. [Attackers](https://www.splunk.com/en_us/blog/learn/threat-actors.html) could use this method to steal sensitive information like login details.  
  
*(Related reading:*[*What is spoofing?*](https://www.splunk.com/en_us/blog/learn/spoofing.html)*)*

## ARP best practices

Despite its major role in facilitating communication between devices, ARP also brings the challenge of ARP spoofing. To avoid and mitigate such issues, all relevant parties working with ARP need to follow best practices and use ARP security appliances and software to [monitor ARP traffic and ensure its security](https://www.splunk.com/en_us/blog/learn/security-monitoring.html).

See an error or have a suggestion? Please let us know by emailing [splunkblogs@cisco.com](mailto:splunkblogs@cisco.com).

*This posting does not necessarily represent Splunk's position, strategies or opinion.*

[](https://www.splunk.com/en_us/form/the-essential-guide-to-network-data.html)

**Essential Guide to Network Data**

Download this free e-book to learn about the basics of network data.

[**Download now**](https://www.splunk.com/en_us/form/the-essential-guide-to-network-data.html)



[**Shanik**](https://www.splunk.com/en_us/blog/author/shanika-wickramasinghe.html)

The **Domain Name System (DNS)** is a hierarchical and decentralized system used to **translate human-readable domain names** (such as **www.example.com**) into **IP addresses** (like **192.168.1.1**) that computers can use to identify each other on the internet or a local network. This process is crucial for enabling the functioning of the internet, as computers and other devices communicate using IP addresses, but humans find it easier to remember domain names.

**Role of DNS:**

1. **Hostname to IP Address Resolution**:
   * The primary role of DNS is to resolve **domain names** (such as **www.example.com**) to **IP addresses** (such as **93.184.216.34**). This allows users to access websites and services using easy-to-remember names instead of numerical IP addresses, which are harder for people to remember.
2. **Decentralized and Hierarchical Structure**:
   * DNS operates through a **distributed database** that is organized in a hierarchical manner. It starts with the root domain, then branches into top-level domains (TLDs) like .com, .org, .net, and country-specific domains like .uk, .us. Below these TLDs, there are second-level domains (like **example.com**), followed by subdomains (like **www.example.com**).
3. **Name Resolution Process**:
   * When a user types a domain name into a web browser, the system sends a query to a **DNS resolver**. The resolver checks its cache to see if it already knows the IP address for the domain. If not, it queries authoritative DNS servers to resolve the name by following the hierarchy until it finds the correct IP address.
4. **Load Balancing and Redundancy**:
   * DNS can also play a role in load balancing by associating multiple IP addresses with a single domain name. This allows traffic to be distributed across multiple servers, improving performance and reliability.
5. **Supporting Internet Functionality**:
   * DNS is essential for the operation of many internet services beyond just web browsing, including email (via MX records), VoIP, and more. It helps route emails to the correct mail servers and ensures smooth communication across the internet.

**In Summary:**

The **Domain Name System (DNS)** is critical for translating domain names into IP addresses, making the internet more user-friendly and ensuring the smooth operation of various online services. Without DNS, users would have to remember numerical IP addresses instead of simple and memorable domain names.

**DNS Definition**

The Domain Name System (DNS) turns domain names into IP addresses, which browsers use to load internet pages. Every device connected to the internet has its own IP address, which is used by other devices to locate the device. DNS servers make it possible for people to input normal words into their browsers, such as Fortinet.com, without having to keep track of the IP address for every website.

**What Is A DNS Server?**

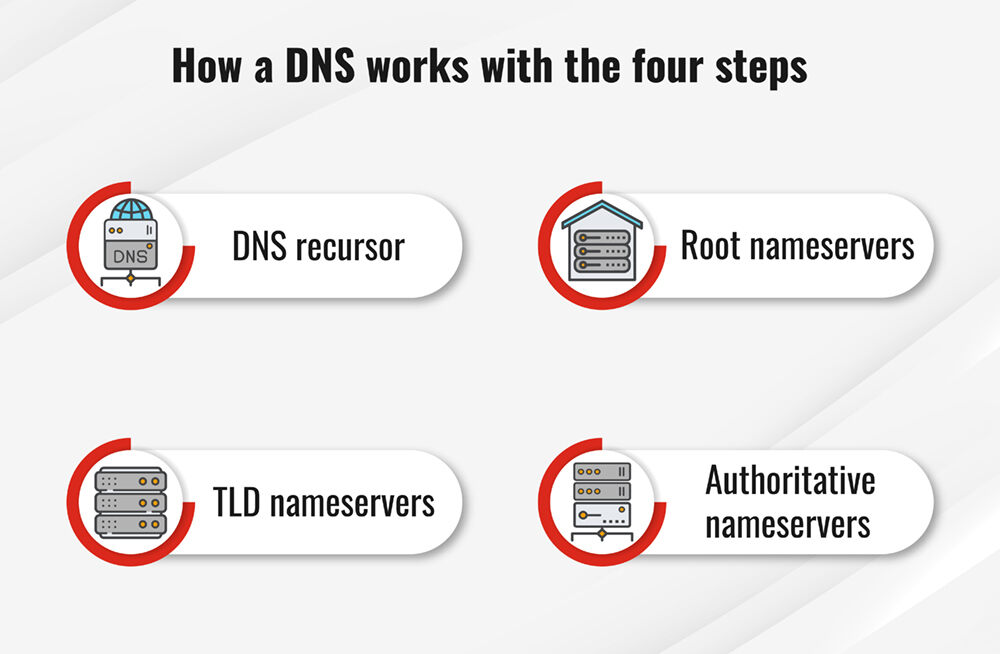
A DNS server is a computer with a database containing the public IP addresses associated with the names of the websites an IP address brings a user to. DNS acts like a phonebook for the internet. Whenever people type domain names, like Fortinet.com or Yahoo.com, into the address bar of web browsers, the DNS finds the right IP address. The site’s IP address is what directs the device to go to the correct place to access the site’s data.

Once the DNS server finds the correct IP address, browsers take the address and use it to send data to content delivery network (CDN) edge servers or origin servers. Once this is done, the information on the website can be accessed by the user. The DNS server starts the process by finding the corresponding IP address for a website’s uniform resource locator (URL).

**How Does DNS Work?**

In a usual DNS query, the URL typed in by the user has to go through four servers for the IP address to be provided. The four servers work with each other to get the correct IP address to the client, and they include:

* 1. **DNS recursor**: The DNS recursor, which is also referred to as a DNS resolver, receives the query from the DNS client. Then it communicates with other DNS servers to find the right IP address. After the resolver retrieves the request from the client, the resolver acts like a client itself. As it does this, it makes queries that get sent to the other three DNS servers: root nameservers, top-level domain (TLD) nameservers, and authoritative nameservers.
  2. **Root nameservers**: The root nameserver is designated for the internet's DNS root zone. Its job is to answer requests sent to it for records in the root zone. It answers requests by sending back a list of the authoritative nameservers that go with the correct TLD.
  3. **TLD nameservers**: A TLD nameserver keeps the IP address of the second-level domain contained within the TLD name. It then releases the website’s IP address and sends the query to the domain’s nameserver.
  4. **Authoritative nameservers**: An authoritative nameserver is what gives you the real answer to your DNS query. There are two types of authoritative nameservers: a master server or primary nameserver and a slave server or secondary nameserver. The master server keeps the original copies of the zone records, while the slave server is an exact copy of the master server. It shares the DNS server load and acts as a backup if the master server fails.



**[[](https://www.fortinet.com/resources/reports/threat-landscape-report?utm_content=cg-cybersecurity-ribbon)](https://www.fortinet.com/resources/reports/threat-landscape-report?utm_content=cg-cybersecurity-ribbon" \t "_self)**

**[2025 Global Threat Landscape Report](https://www.fortinet.com/resources/reports/threat-landscape-report?utm_content=cg-cybersecurity-ribbon" \t "_self)**

[Use this report to understand the latest attacker tactics, assess your exposure, and prioritize action before the next exploit hits your environment.](https://www.fortinet.com/resources/reports/threat-landscape-report?utm_content=cg-cybersecurity-ribbon" \t "_self)

**[Download Now](https://www.fortinet.com/resources/reports/threat-landscape-report?utm_content=cg-cybersecurity-ribbon" \t "_self)**

**Authoritative DNS Servers vs Recursive DNS Servers: What’s The Difference?**

Authoritative nameservers keep information of the DNS records. A recursive server acts as a middleman, positioned between the authoritative server and the end-user. To reach the nameserver, the recursive server has to “recurse” through the DNS tree to access the domain’s records.

**Authoritative DNS Server**

To use the phone book analogy, think of the IP address as the phone number and the person’s name as the website’s URL. Authoritative DNS servers have a copy of the “phone book” that connects these IP addresses with their corresponding domain names. They provide answers to the queries sent by recursive DNS nameservers, providing information on where to find specific websites. The answers provided have the IP addresses of the domains involved in the query.

Authoritative DNS servers are responsible for specific regions, such as a country, an organization, or a local area. Regardless of which region is covered, an authoritative DNS server does two important jobs. First, the server keeps lists of domain names and the IP addresses that go with them. Next, the server responds to requests from the recursive DNS server regarding the IP address that corresponds with a domain name.

Once the recursive DNS server gets the answer, it sends that information back to the computer that requested it. The computer then uses that information to connect to the IP address, and the user gets to see the website.

**Recursive DNS Server**

After a user types in a URL in their web browser, that URL is given to the recursive DNS server. The recursive DNS server then examines its [**cache memory**](https://www.fortinet.com/resources/cyberglossary/what-is-caching) to see whether the IP address for the URL is already stored. If the IP address information already exists, the recursive DNS server will send the IP address to the browser. The user is then able to see the website for which they typed in the URL.

On the other hand, if the recursive DNS server does not find the IP address when it searches its memory, it will proceed through the process of getting the IP address for the user. The recursive DNS server's next step is to store the IP address for a specific amount of time. This period of time is defined by the person who owns the domain using a setting referred to as time to live (TTL).

**DNS Servers And IP Addresses**

Computers and various devices that use the internet depend on IP addresses to send a user's request to the website they are attempting to reach. Without DNS, you would have to keep track of the IP addresses of all the websites you visit, similar to carrying around a phone book of websites all the time. The DNS server allows you to type in the name of the website. It then goes out and gets the right IP address for you. Armed with the IP address, your computer (or browser) can bring you to the site.

For instance, if you input www.fortinet.com in your web browser, that URL, on its own, cannot bring you to the website. Those letters cannot be “read” by the servers that connect you with the site. However, the servers are able to read IP addresses. The DNS server figures out which IP address corresponds with [**www.fortinet.com**](https://www.fortinet.com/) and sends it to your browser. Then the website appears on your device’s screen because the browser now knows where to take your device.

**DNS Server Not Responding? What Does That Mean?**

You may get a message that says “DNS server isn’t responding” after entering a domain name in the URL bar of your browser. This means there was an attempt to communicate with the DNS server, but the server failed to return a result. This could be due to a few different things:

* 1. Your internet connection is weak or unstable, making it hard for your browser to communicate with the DNS server
  2. Your DNS settings or browser need to be updated
  3. There is an issue with the DNS server, such as a loss of power at the data center where it is housed

**Best DNS Servers**

Here are some of the top DNS servers available:

* 1. **Cloudflare 1.1.1.1.**This is a simple-to-use DNS service that comes with tutorials for all of the most popular operating systems, such as Mac, Windows, Android, iOS, and Linux. Users can also use Cloudflare’s service to block adult content.
  2. **Google Public DNS.** The Google Public DNS service is different from Cloudflare’s in that it is designed for more technically adept users. But you can find tutorials if needed.
  3. **Quad9.** Quad9’s DNS service is renowned for its fast performance. It also claims to block malicious sites using threat intelligence data.

**Browser DNS Caching**

The operating system (OS) used by your device stores DNS resource records through the use of caching. Caching prevents redundancy when someone tries to go to a site. This, in turn, reduces the amount of time it takes to get to the website. If the device you are using recently went to the page it is trying to access, the IP address can be supplied by the cache. In this way, the website request can be completed without involving the DNS server.

The DNS cache, therefore, helps streamline the DNS lookup process that would otherwise be necessary to link a domain name to an IP address. This makes the process of getting to the website much faster.

**OS DNS Caching**

The operating systems of many devices are capable of maintaining a local copy of DNS lookups. This makes it possible for the OS to quickly get the information it needs to resolve the URL to the correct IP address.

**How To Perform A DNS Lookup**

Each domain has DNS records, and these are pulled by nameservers. You can check the status of the DNS records associated with your domain. You can also examine the nameservers to ascertain which records are being pulled by the servers. On a Windows computer, for example, this is done using the NSLOOKUP command. Here’s how to do it:

* 1. Access the Windows command prompt by going to Start >> command prompt. You can also get to it via Run >> CMD.
  2. Type NSLOOKUP and then hit Enter. The default server gets set to your local DNS, and the address will be your local IP address.
  3. You then set the type of DNS record you want to look up by typing "set type=##" where "##" is the record type, then hit Enter. You can also use A, AAAA, A+AAAA, ANY, CNAME, MX, NS, PTR, SOA, or SRV as the record type.
  4. Enter the domain name you want to query. Hit Enter.
  5. At this point, the NSLOOKUP returns the record entries for the domain you entered.

**What is a DNS resolver?**

A DNS resolver is also referred to as a recursive resolver. It is designed to take DNS queries sent by web browsers and applications. The resolver receives the website URL, and it then retrieves the IP address that goes with that URL.

**What are the types of DNS queries?**

During the DNS lookup process, three different kinds of queries are performed. The queries are combined to optimize the resolution of the DNS, saving time.

* 1. Recursive query
  2. Iterative query
  3. Non-recursive query

**Free vs Paid DNS Servers: What Is The Difference?**

In some cases, a regular user may not need a paid DNS server. However, there are significant benefits of paying for a premium DNS.

* 1. [**Dynamic DNS (DDNS)**](https://www.fortinet.com/resources/cyberglossary/dynamic-dns): A DDNS maps internet domains, matching them to IP addresses. This enables you to get into your home computer no matter where you are in the world. DDNS is different from a regular DNS because it works with changing or dynamic IP addresses, making them a good choice for home networks.
  2. **Secondary DNS**: A secondary DNS nameserver makes sure that your domain does not go offline. It provides you with a redundancy or backup that can be accessed in the event of a complication.
  3. **Management interface**: Many paid DNS servers offer users a dashboard they can use to manage their service and tweak it according to their needs.
  4. **Two-factor authentication**: You can provide protection for your domain with an extra level of authentication.
  5. **More security**: When you make use of a paid DNS server, you get another protective level of security. This helps shield your website from attackers.
  6. **Better, faster performance**: A paid DNS server comes with a service-level agreement (SLA). Each SLA guarantees a high rate of DNS resolution, often between 99% and 100%.
  7. **Customer service**: With a paid DNS server, you get the additional advantage of customer service that can answer questions and troubleshoot any issues.

**What is DNS cache poisoning?**

DNS cache poisoning, also called DNS spoofing, involves the introduction of corrupt DNS data into the resolving device’s cache. This results in the nameserver returning the wrong IP address.

The operating systems of many devices are capable of maintaining a local copy of DNS lookups. This makes it possible for the OS to quickly get the information it needs to resolve the URL to the correct IP address.

**DNS FAQs**

**What is DNS?**

**What is an example of DNS?**

**How do I find my DNS?**

**What are the types of DNS?**

**Is changing DNS safe?**

**Should I use private DNS?**

A **Virtual Local Area Network (VLAN)** is a logical partition within a physical network that groups together a set of devices, regardless of their physical location. The main purpose of a VLAN is to improve network management, security, and performance by segmenting a larger network into smaller, more manageable groups.

**Purpose of a VLAN:**

1. **Network Segmentation**:
   * VLANs allow the division of a single physical network into multiple **logical networks**. Devices within the same VLAN can communicate with each other as if they are on the same physical network, regardless of their actual physical locations. This segmentation helps organize the network based on function, department, or security requirements.
2. **Improved Security**:
   * By creating VLANs, network traffic can be isolated between different groups of users or devices. For example, a **finance department** can be placed in its own VLAN, preventing unauthorized access from other departments (like HR or Marketing). This reduces the risk of security breaches and limits the spread of malicious traffic.
3. **Enhanced Performance**:
   * VLANs reduce **broadcast traffic** within the network. Since broadcast messages are confined to devices within the same VLAN, this prevents unnecessary traffic from reaching devices outside of the VLAN, improving network performance and reducing congestion.
4. **Simplified Network Management**:
   * VLANs enable easier management of large networks. Changes such as adding, removing, or moving devices can be done through the VLAN configuration rather than physically re-cabling or moving devices around. VLANs also support policies such as Quality of Service (QoS) and traffic prioritization, allowing network administrators to optimize traffic flow.
5. **Efficient Use of Resources**:
   * VLANs allow for better utilization of network resources by logically grouping devices based on factors like department, function, or service type. This means that resources like network bandwidth and IP address spaces can be allocated more effectively.

**In Summary:**

The purpose of a **VLAN** is to **segregate** network traffic, **enhance security**, **improve performance**, and simplify network management by logically grouping devices based on criteria such as function, department, or security level, regardless of their physical locations in the network.

# LAN (virtual LAN)

By

* [**Terry Slattery,**](https://www.techtarget.com/contributor/Terry-Slattery)NetCraftsmen
* [**John Burke,**](https://www.techtarget.com/contributor/John-Burke)Nemertes Research

### What is a VLAN (virtual LAN)?

A virtual LAN (VLAN) is a logical overlay network that groups together a subset of devices that share a physical LAN, isolating the traffic for each group.

A LAN is a group of computers or other devices in the same place -- e.g., the same building or campus -- that share the same physical network. A LAN is usually associated with an Ethernet ([Layer 2](https://www.techtarget.com/searchnetworking/definition/Data-Link-layer)) broadcast domain, which is the set of network devices an [Ethernet](https://www.techtarget.com/searchnetworking/definition/Ethernet) broadcast packet can reach.

Computers on the LAN connect to the same [network switch](https://www.techtarget.com/searchnetworking/definition/switch), either directly or through wireless access points (APs) connected to the same switch. Computers can also connect to one of a set of interconnected switches, such as a set of access switches that all connect up to a backbone switch. Once traffic crosses a router and engages [Layer 3](https://www.techtarget.com/searchnetworking/definition/Network-layer) (IP-related) functions, it is not considered to be on the same LAN, even if everything stays in the same building or floor. As a result, a location could have many interconnected LANs.

A VLAN, like the LAN it sits atop, operates at Layer 2 of the network, the Ethernet level. VLANs partition a single switched network into a set of overlaid [virtual networks](https://www.techtarget.com/searchnetworking/definition/virtual-networking) that can meet different functional and security requirements. This partitioning avoids the need to have multiple, distinct physical networks for different use cases.

**1. Function of the Internet Control Message Protocol (ICMP):**

**ICMP** is a network layer protocol that is used primarily for **error reporting** and **diagnostic functions** in a network. It is integral to the operation of IP (Internet Protocol) networks. The most common use of ICMP is in the **ping** command, which helps test the reachability of a device on a network.

**Functions of ICMP:**

* **Error Reporting**: ICMP is used by routers and devices to send error messages indicating problems such as network congestion, unreachable destinations, or time-exceeded errors.
* **Diagnostic Tools**: ICMP enables network diagnostic tools such as **ping** and **traceroute**. The **ping** tool sends ICMP Echo Request messages to a destination and listens for ICMP Echo Reply messages to confirm whether the destination is reachable. **Traceroute** uses ICMP to trace the route packets take through a network, identifying each router in the path.
* **Time Exceeded**: ICMP can notify when a packet's Time-to-Live (TTL) has expired, which helps identify routing loops or packet delay problems.

**2. Functions of a DHCP Server:**

A **DHCP (Dynamic Host Configuration Protocol)** server is responsible for dynamically assigning **IP addresses** and other network configuration settings to devices (clients) on a network. DHCP automates the process of IP address allocation, reducing manual configuration and potential conflicts.

**Functions of a DHCP Server:**

* **Assigning IP Addresses**: DHCP dynamically assigns unique IP addresses to devices as they join the network. This eliminates the need for static IP addressing and helps prevent IP address conflicts.
* **Lease Management**: DHCP assigns IP addresses on a temporary basis (known as a lease), typically for a specified period. After the lease expires, the IP address can be reassigned to another device.
* **Providing Network Configuration Information**: In addition to IP addresses, DHCP provides other important configuration settings, including:
  + **Subnet Mask**: Defines the network and host portions of an IP address.
  + **Default Gateway**: Specifies the IP address of the device that routes traffic outside the local network.
  + **DNS Servers**: Specifies the addresses of DNS servers for name resolution.
* **Automating IP Management**: With DHCP, network administrators don’t need to manually assign IP addresses to each device, reducing administrative overhead and the risk of errors.

**3. Use of NAT (Network Address Translation):**

**NAT (Network Address Translation)** is a technique used in networking to modify the source or destination IP address of packets as they pass through a router or firewall. It is commonly used in home and enterprise networks to enable multiple devices to share a single public IP address when accessing the internet.

**Functions of NAT:**

* **Private to Public IP Translation**: NAT allows devices with private IP addresses (e.g., 192.168.x.x, 10.x.x.x) within a local network to access external resources (like websites) by translating their private IP addresses into a single public IP address.
* **Conserving IP Addresses**: NAT helps conserve public IP addresses, which are a limited resource. By allowing many devices on a private network to share one public IP, NAT reduces the number of public IPs needed.
* **Enhancing Security**: NAT acts as a basic firewall by hiding internal network addresses. External devices cannot directly access devices within the private network, thus providing a layer of security.
* **Port Address Translation (PAT)**: A variation of NAT, known as **Port Address Translation** (PAT), allows multiple devices to share a single public IP address using different port numbers to distinguish between connections.

**4. Function of SMTP (Simple Mail Transfer Protocol):**

**SMTP** is a protocol used for sending **email messages** between servers and clients in a **client-server** architecture. SMTP is responsible for the **outgoing email** delivery, ensuring that emails are transferred from the sender’s mail server to the recipient’s mail server.

**Functions of SMTP:**

* **Sending Emails**: SMTP is used to send messages from the sender's email client (like Outlook or Gmail) to the sender's mail server, and then from the sender's mail server to the recipient's mail server.
* **Routing Email Messages**: SMTP helps route email through different mail servers until it reaches the destination email server.
* **Ensuring Reliable Delivery**: SMTP works with other protocols like **IMAP (Internet Message Access Protocol)** or **POP3 (Post Office Protocol)** to retrieve and store messages on the client’s email application, but SMTP handles the actual sending of emails.
* **Simple and Efficient**: SMTP works in a simple request-response manner to push messages to the next step in the delivery process, but it is not used for retrieving emails from the server (IMAP or POP3 handles that part).

**5. Advantages of Using SSH (Secure Shell):**

**SSH (Secure Shell)** is a cryptographic network protocol used for securely accessing and managing devices over an unsecured network. It replaces older protocols like **Telnet** and **rlogin**, providing a much higher level of security.

**Advantages of SSH:**

* **Encrypted Communication**: SSH ensures that all data sent over the network is encrypted, protecting it from eavesdropping and man-in-the-middle attacks.
* **Secure Remote Access**: SSH allows system administrators to securely log into remote servers and manage them from anywhere, even over public networks (e.g., the internet).
* **Authentication**: SSH uses various methods of authentication, such as passwords or SSH keys, to verify the identity of the user before granting access to the remote system. SSH keys provide a more secure and convenient authentication method compared to passwords.
* **Data Integrity**: SSH includes mechanisms to ensure data integrity by using checksums to verify that the data has not been altered during transmission.
* **Secure File Transfers**: SSH supports secure file transfer protocols, such as **SFTP (Secure File Transfer Protocol)** and **SCP (Secure Copy Protocol)**, for transferring files between machines securely.
* **Tunneling and Port Forwarding**: SSH can create secure tunnels and port forwarding to secure data traffic that would otherwise be insecure, allowing for encrypted communication over untrusted networks.

**Summary:**

* **ICMP** is used for error reporting and diagnostics in a network (e.g., with tools like **ping**).
* A **DHCP server** dynamically assigns IP addresses and provides network configuration information to devices on a network.
* **NAT** enables multiple devices within a local network to share a single public IP address when accessing the internet and helps with security and IP address conservation.
* **SMTP** is responsible for sending email messages between servers and clients over a network.
* **SSH** provides secure remote access to servers and encrypts communication, offering advantages such as authentication, data integrity, and secure file transfer.