\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Configuring a Wireless LAN Controller

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Author*:

Gabriel Rosas



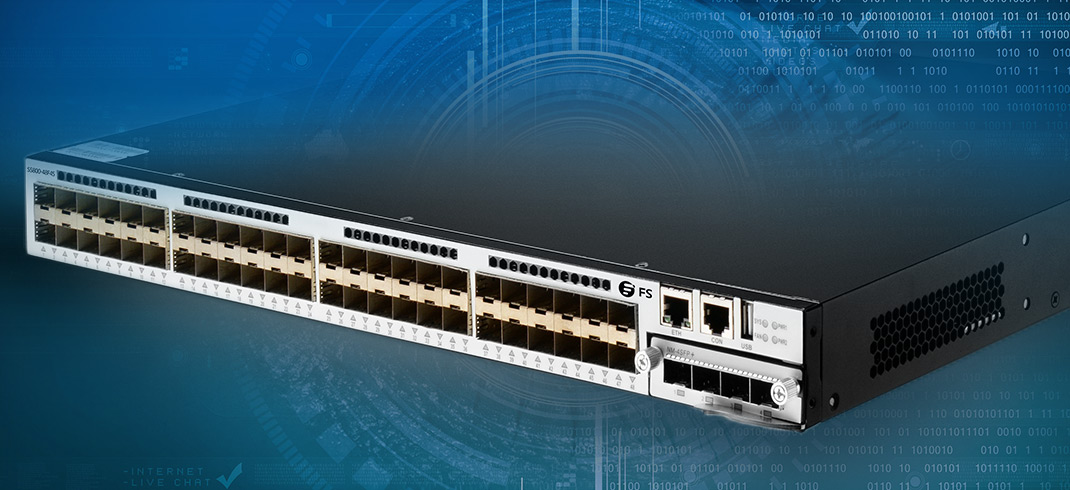
Purpose

As the demand for WiFi access grows, more Wireless Access Points are thrust into larger networks to ensure the signal extends the desired range. However, each additional access point brings more configuration an administrator must account for, becoming increasingly tedious as the number increases. Wireless LAN Controllers monitor and manage access points in bulk, removing a lot of extra hassle for network engineers. In this Lab, I attempt to host a wireless network with access to the internet using one of these controllers.

Background Information

It should be noted that my methods and understanding may not be conventional, as my only goal was to establish connection to the internet for wireless clients using a Wireless LAN Controller. I did not consider security, though that may be something I may implement in the future. That said, in order to understand how each part fits together, I will attempt to cover the following necessary concepts:

* *Switching*
* *VLANs*
* *DHCP*
* *Wireless*
* *The Cisco 5500 series Wireless LAN Controller*
* *NAT*

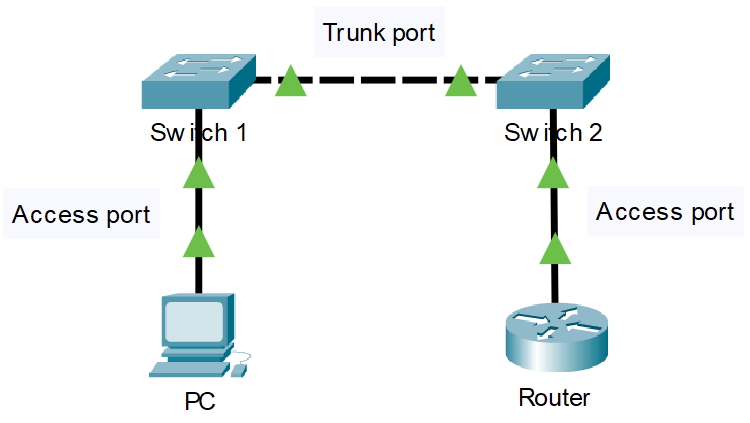


How does a Switch function?

A simple way to think of a switch is like a power strip: a single outlet in the wall can be split into multiple outlets if needs be. While a power strip extends the number of *outlets*, a switch extends the number of *ethernet ports*. A typical router only has two or three ethernet ports, so at times when more are needed, a network administrator would introduce a switch.

Each port on the switch stores the MAC address of the connected device. Instead of relying on broadcasts, switches intelligently direct traffic to the correct destination based on the MAC address. The **Content-Addressable Memory (CAM) table** maps each *port* to a *MAC address* so the switch can easily direct data using the destination MAC address of a packet through the correct port.

If a host connected to the *Fast Ethernet 0/1* port requests a webpage, data will route to and from the webserver and eventually back to the switch (though it will probably take mere milliseconds). After reaching the switch, the packets forward through the port associated with their destination MAC address. Switches can be joined together to support even more ports. A switch-to-switch connection is known as a **trunk port**. Otherwise, the connection is an **access port**. These port types will bring more significance when we look at VLANs.



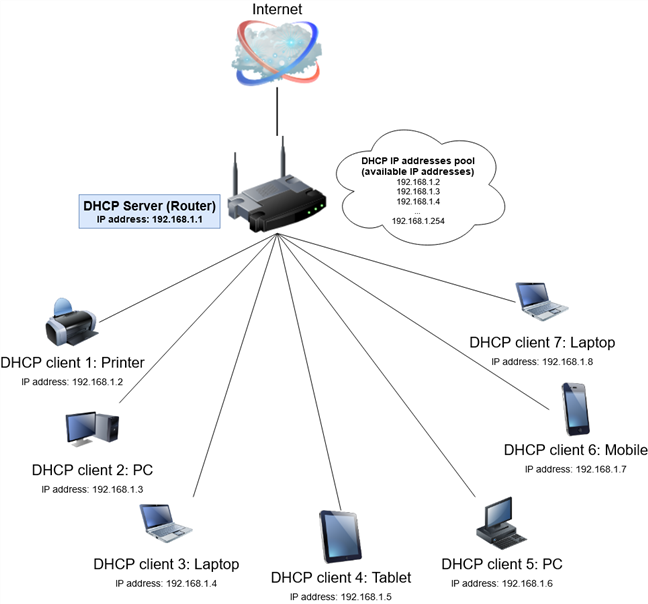
What are VLANs?

Imagine a square room built entirely out of doors. Anyone can pass through a door and leave through another since there is no interference, no rules dictating otherwise. Now, consider a solid wall placed to divide that room into two sections. While the outer foundation is still all doors, the inside is completely split with no ability to communicate with the other section. In essence, this is what you can do to a switch. You can section ports on a switch into a certain VLAN and those ports will only be able to communicate with their respective VLAN. So even if two devices were plugged in directly next to each other, they could only relay traffic if they were on the **same** VLAN. But as with every rule, there is the exception: **trunk** ports can relay traffic of **multiple** VLANs on the **same** port.

A major use case for VLANs is separating sensitive traffic, perhaps confidential data, on a network from the general traffic. A business might have multiple departments, each with little reason to communicate with the other, and therefore separate them with different VLANs. Having more VLANs also reduces the size of the broadcast domain – the number of IPs on a subnet – granting less packet congestion.

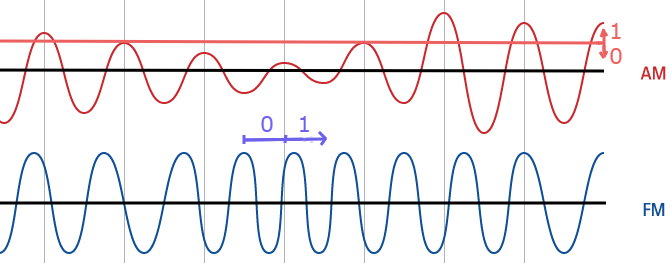
Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a protocol, configurable on a server, that leases IP addresses to clients automatically. For the average person with little networking experience, dynamically receiving addresses is much less confusing that statically assigning them. When many devices need addresses, DHCP provides them, leasing zero duplicates.



How does wireless work?

All wireless transmissions are made possible by radio waves. Radio signals can produce different amplitudes or frequencies of waves, which can be converted to binary 1s and 0s based on how large the amplitude or frequency of the wave is. Imagine you drop a stone into a still lake. Ripples begin to emanate from where the stone hit the water, some big and some small. Almost like a hidden code, we can deduce whether a wave represents a “0” or a “1”, depending on if that wave is above a certain size (Amplitude Modulation) or how compact the waves are together (Frequency Modulation).



Radio waves used for WiFi communication are very similar to those used for walkie-talkies, cell phones, even clinical MRI devices – the difference is the frequency range allocated for each protocol. WiFi transmits at frequencies of roughly 2.4 GHz or 5 GHz. The higher a frequency, the more data a signal can carry.

WiFi uses 802.11 networking standards. Here are some of the 802.11 protocols that have evolved over the decades (From worst to best).

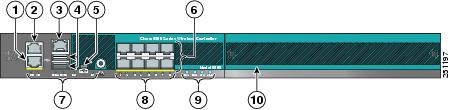
* 802.11b
  + Introduced in 1999
  + Transmits at 2.4 GHz
  + Can handle up to 11 Mbps
  + Cheap cost made it popular, but is less common now as faster standards became cheaper
* 802.11a
  + Introduced after 802.11b
  + Transmits at 5 GHz
  + Can handle up to 54 Mbps
  + Utilizes OFDM, a more efficient coding technique that spits a radio signal into several sub-signals, greatly reducing interference
* 802.11g
  + Released in 2003
  + Transmits at 2.4 GHz
  + Can handle up to 54 Mbps
  + Uses the same OFDM coding as 802.11a
* 802.11n
  + Introduced in 2009
  + Transmits at 2.4 or 5 GHz
  + Can handle up to 140 Mbps
  + Can transmit up to 4 streams of data
* 802.11ac
  + Introduced in 2014
  + Transmits at 5 GHz
  + Can handle up to 450 Mbps
  + Backwards compatible, much less prone to interference and far faster than its predecessors
* 802.11ax (WiFi 6)
  + Introduced in 2019
  + Transmits at 2.4 or 5 GHz
  + Can handle up to 9.2 Gbps
  + Allows manufacturers to install more antennas on a router, accepting more connections at once without worries of interference
* 802.11be (WiFi 7)
  + Projected to be the standard by 2024
  + Transmits at 2.4, 5, or 6 GHz
  + Rates as high as 40 Gbps

Cisco 5508 Wireless LAN Controller

As I mentioned in my introduction, Wireless LAN Controllers (WLCs) are used to remotely manage access points. For those unfamiliar, access points are devices that host wireless networks, allowing clients to join a network without any physical link. Access points usually connect to routers or switches, which handle all the routing and necessary steps to ensure connectivity to the internet. Access points will either run *lightweight* or *autonomous* operating systems. Lightweight access points are controlled by a WLC, and autonomous ones are standalone. Since I am using a WLC, the access points are lightweight.

Below are the physical and logical topologies of the 5508 WLC. It is useful to understand the ports and interfaces on the device before configuring it. The WLC operating system is release 8.3.122.0.

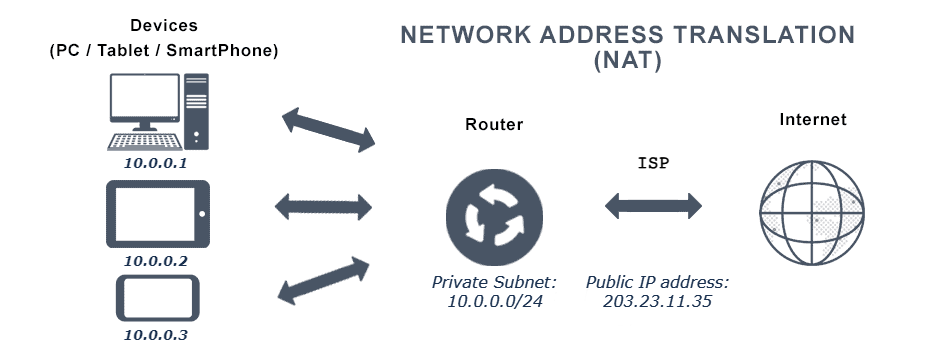




1. Redundant port (RJ-45)
2. Service port (RJ-45)
   * *For remotely controlling and managing the WLC using a secure connection*
   * *The service port’s IP must be on a different subnet from the management interface*
3. Console port (RJ-45)
   * *For serial connections*
4. USB ports 0 and 1 (Type A)
5. Console port (Mini USB Type B)
6. SFP distribution ports 1-8
   * *SFP (Small Form-factor Pluggable) ports allow fiber or copper connections depending on the module plugged in to the port*
7. Management port LEDs
8. SFP port LEDs
9. Alarm and Notification LEDs
10. Expansion module slot

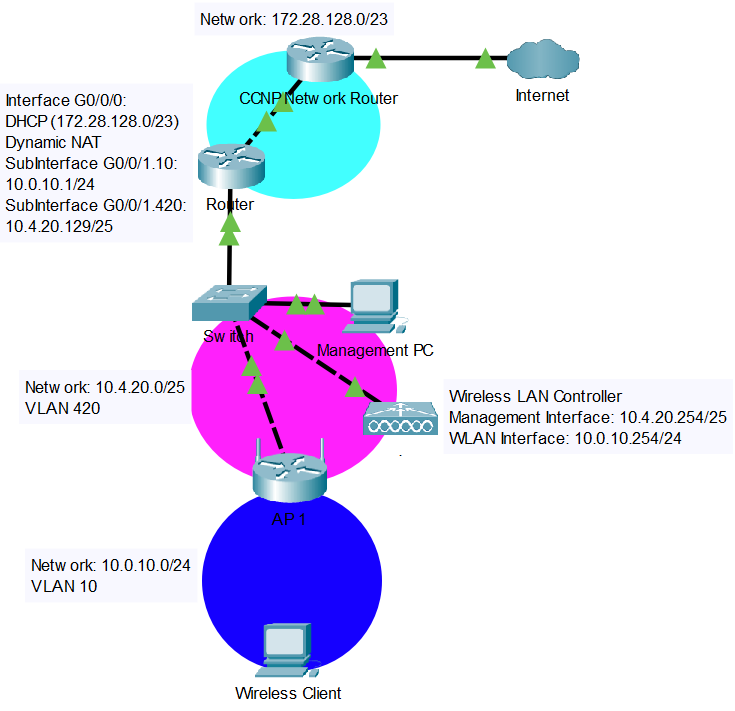
Network Address Translation

Network Address Translation (NAT) is a process that enables a unique IP address to represent an entire group of devices. Typically, this IP address is “Public”, administered by one’s ISP.



In this example, the public IP, *203.23.11.35*, is representing the *10.0.0.0/24* subnet. There can be multiple networks that use the same private subnet – what matters is that the public IP is unique. With so many devices nowadays, the total IPv4 address pool is running out. NAT helps conserve these addresses by translating multiple private IPs through one public IP. This is known as dynamic NAT. There are other kinds of NAT such as static or PAT, but for this Lab I use dynamic NAT.

Network Diagram



Process

In every lab that I have previously done, there were times when I was overwhelmed due to the complexity or size of what I had to do. Usually, this was my fault, as I always tried to brute-force my way through a problem, biting off more that I could chew, instead of breaking it down into practical chunks. Since I was diving into a realm where I had little experience – working with wireless and the WLC – I decided to create manageable goals for each day, to prevent becoming helplessly lost. While some goals spanned more than a day, they helped narrow my focus so I could be more productive.

Connecting to the WLC

I began with arguably the most important step in working with any device: connecting to it. All I wanted was to access an interface which lets me interact with the WLC. Since Cisco routers and Switches are mainly configured from console, I connected my computer to the WLC with a console cable. Almost immediately, I was met with a bunch of questions regarding the initial setup. What should the system name, username, or password be? Do I want to enable SNMP? Enter the service-port IP address. Overwhelmed by the options, I took a step back and searched for some documentation.

Initial Setup

The documentation wasn’t very hard to find. It did, however, make up for that by being as overwhelming as the questions. Luckily, there was a section on initial setup. Except the steps were not using the console interface, they were using the web interface. I decided to primarily configure the WLC using the web interface.

The documentation provided me with a lot of options, though many of which were not expanded upon to my liking. To flush out the details that were not clear to me, I decided to compile my own list of the most important information, providing extra reference to areas that lacked explanation, so I could refer to that instead of the official documentation. The major steps that I compiled are as followed: the system name is the WLC’s name, the username and passwords are for administrative login; leave SNMP default, it probably doesn’t need to change; the service port is for remote management and must be on a different subnet from the management interface; enable Link-Aggregation (it apparently simplifies controller configurations); use *VLAN 0*, assign *port 1* and configure a separate DHCP server for the management interface; leave the RF group as default; use the country code that the network will be deployed in, multiple can be selected if the network extends across multiple countries; use a *192.0.2.0/24* address for the virtual interface since this is for testing purposes (*RFC 5727*); the wireless profile and SSIDs will show up as the WiFi networks; disable the RADIUS server and enable all *802.11* protocols. While I ended up changing some settings later on, taking the time in the beginning to create this configuration guide massively propelled me forward. Referencing my guide, I made a list of configurations ahead of time before finishing the setup wizard.

Connecting an Access Point to the WLC

After completing the wizard with my researched configurations, I finally had access to the main control panel of the WLC. Although the documentation was significantly helpful for the initial setup, I decided to use other sources for the rest; it was difficult picking the details that I needed out of a thousand-page PDF file.

To establish a connection with a WLC, an access point needs an IP address. In lightweight mode, it is conventional for the access point to gain the information by DHCP. There may have been a way to set up DHCP internally on the WLC, however, I chose to use an external Cisco router as a DHCP server. The access point received addresses, but then complained because it didn’t know the address of the controller. Fortunately, this problem is solvable with DHCP option 43, a configurable setting that forwards WLC addresses in hexadecimal. With DHCP option 43 configured, everything should have been working. But it wasn’t.

To top it all off, my PC stopped receiving DHCP information. While the timing may seem coincidental with the new DHCP server changes, the error raised, “*an error occurred while renewing interface Ethernet: The object already exists*”, had no relation to DHCP option 43. I tried a couple online solutions: resetting device interfaces, resetting Winsock, and restarting my computer. Not even relating to the WLC, this problem took me some time to figure out. Eventually I went into Windows *adapter settings* and noticed VirtualBox, a VMware interface, was running. Out of ideas, I disabled everything that was not ethernet. Thankfully, the virtual interface was the problem, and I was able to resume troubleshooting the WLC.

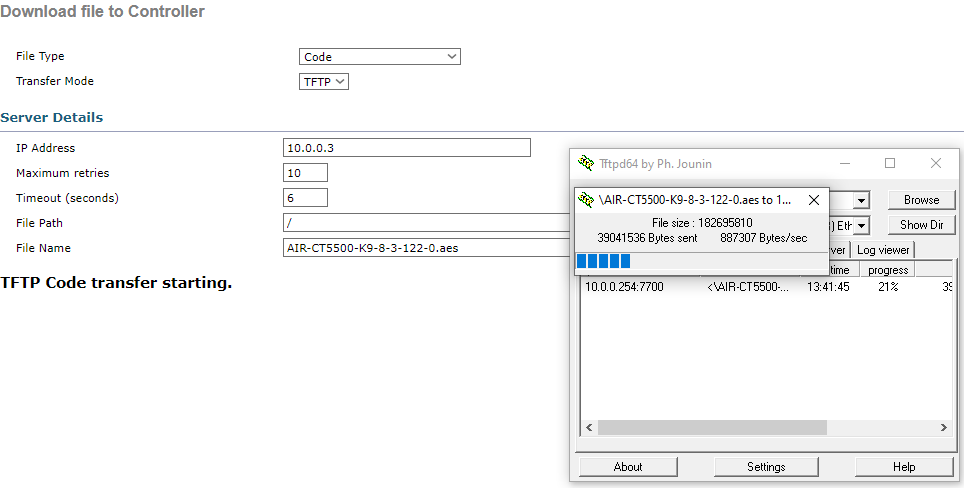
Assuming the DHCP server was properly configured, there were only two devices which could be causing issues: the WLC or access point. After viewing debug logs of both devices, I decided to start clean and reset the access point to factory default. The WLC had just been initialized from factory, so it was far more likely something was incorrectly configured on the access point. My first approach was clearing the CAPWAP configuration, the protocol used for a connection between access points and WLCs, which achieved nothing. Then there was the “complete wipe” approach. For this to work, I needed to enter ROM monitor (ROMMON) – low-level firmware that runs before the operating system boots – by holding down the power button then type *delete flash:private-multiple-fs*, deleting the configuration file. Once the old configuration file is removed from ROMMON, the access point will create a brand-new one on boot.

Resetting the access point to factory default did not fix any errors, oddly, it instead increased the log messages. These new log messages sent me down a new path – now they were claiming incompatibility between the WLC IOS and the access point IOS. The integrated operating system (IOS) on the access point was relatively new, which could only mean one thing. The WLC was outdated. This gave me two options: downgrade the access point IOS or upgrade the WLC IOS. But before I could work on that, I was hit with another problem.

The Cisco 5508 Wireless LAN Controller was released on the 18th of May 2009. How could this possibly be related? Well, access points and WLCs both require in-date Manufacturer-Installed Certificates (MIC) to establish a secure *DTLS-tunnel* between them for encrypting CAPWAP control traffic. These certificates expire ten years after the manufacturing date. As of configuring the WLC in early December 2021, the certificate had long expired. Now I had two problems: installing a new operating system and figuring out a solution for the MIC.

There was a workaround to ignore the MIC: running the command *config ap cert-expiry-ignore mic enable*. However, the current IOS, release *7.0.112.21*, on the WLC did not support that command. I could tackle both these problems at once, but that came at the cost of upgrading the IOS of the WLC. The reason for this being costly is quite literal – obtaining a newer IOS requires a license.

Eventually, I obtained the IOS with the help of a friend. I updated the controller, ran the command above, and marveled as the access point appeared in the list.





Setting up VLANs

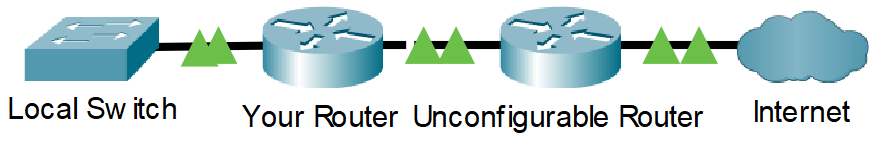
Once the links established, my first order of business was to change the topology. After spending a lot of time troubleshooting and interacting with the WLC, I built up insight on how the WLC operates. I needed to correct some issues in my topology, such as introducing different VLANs for the management and user networks. I started by configuring the web interface, but made a mistake, loosing connectivity. I think I mistyped an IP. This meant was I had to switch to the console interface instead.

The console interface was much like any other Cisco console interface. Once connected, I found some commands online to change the management interface’s IP and VLAN. I set up VLANs on a switch to accompany the new WLC configurations and added sub-interfaces on my router to handle the new VLANs on the switch.

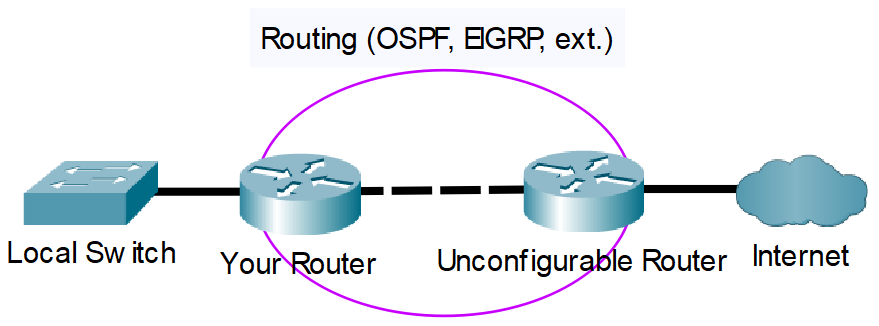
Each interface of the WLC is mapped to a Wireless LAN. Any traffic that a WLAN receives is sent out its respective interface. During the mapping process, I had a problem where a configured WLAN would not appear as a mappable option on the interface that I wanted it mapped to. Searching the configurations, I discovered that *dynamic AP management*, the setting for the management interface, was wrongly enabled on the user-wlan interface. Essentially, this meant that management was being done within the same domain as the clients. Bad. Simply swapping the *dynamic* *AP management* setting to the correct interface solved the problem. At this point, any client could join the wireless network but could not access the internet.

Configuring NAT

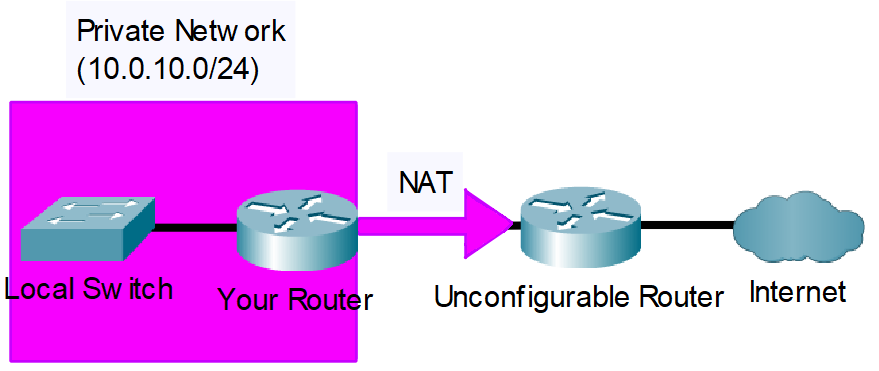
The only reason I needed to configure dynamic NAT was because there was a router blocking the path between my local network and the internet. Consider this topology.



In this topology, the unconfigurable router has access and routes to the internet but doesn’t have any routes to the local network, therefore, isolating the local network from the internet. My initial idea was to set up some form of routing, perhaps using OSPF or EIGRP, between the two routers.



However, it was problematic setting up a routing protocol since I needed access to both routers. One of them was inaccessible. Luckily, the unconfigurable router had an interface connecting to my local network. Since the local router was directly connected to the unconfigurable router, that meant it had connectivity to the internet on the directly connected interface. Using dynamic NAT, I forwarded the local traffic to the unconfigurable router, bypassing the routing. I suppose one could think of NAT like a mayor, speaking for his people.



After NAT was configured, I could hop on to my wireless network and browse YouTube.

Lab Commands

|  |  |
| --- | --- |
| **command** | A statement necessary for a configuration to work, denoted in bold |
| **[*argument*]** | An argument necessary for a command to function, denoted in bold italics. |
| *optional-statement*  *<optional argument>* | An optional argument or statement, not necessary for a command to function, denoted in italics |

// IPv4 DHCP Configuration

Router(config)# **ip dhcp exclude-address [*initial ip*]** <*final ip*>

* Set an IP or range of IPs to exclude from the pool

If the network administrator so chooses to exclude a range of IP addresses, the range would be from the *Initial IP* to the *End IP*, inclusive. The *End IP* argument is not necessary when excluding only one IP. Excludes are typically reserved for pre-configured static IP addresses, for example, interfaces on the router.

Router(config)# **ip dhcp pool [*pool name*]**

* Creates a pool for distributing routing information

Dynamic Host Configuration Protocol (DHCP) automates the assignment of IP addresses to devices on the local network. A DHCP pool is used to define the range of IP addresses that the server will divvy out to clients.

Router(dhcp-config)# **network [*network address*] [*subnet mask*]**

* Configures a pool that distributes the specified subnet

Router(dhcp-config)# **default-router [*ip*]**

* Sends the specified default gateway to clients

Router(dhcp-config)# **dns-server [*ip*]**

* Sends the specified DNS server to clients

Router(dhcp-config)# **option 43 hex [*value*]**

* Advertises Wireless LAN Controller(s) addresses in hexadecimal

// VLAN Configuration

Switch(config)# **vlan [*id*]**

* Create a vlan with specified *id*

A Virtual Lan (vlan) is used to partition groups of devices on a switched network.

Switch(config-if)# **switchport mode access**

* Configure an interface to be in access mode

Only one vlan is permitted across an access mode interface. Devices connected to an access-configured interface are usually end devices, such as clients.

Switch(config-if)# **switchport access vlan [*id*]**

* Configure an interface to be part of a specified vlan

To configure the same command access across multiple interfaces, use the command **interface range [*interface*] [*start-end id*]**.

Switch(config-if)# **switchport trunk encapsulation dot1q**

* Configures the interface to use IEEE 802.1Q encapsulation on frames when the interface is configured as a trunk

The user must define the encapsulation before setting an interface as a trunk.

Switch(config-if)# **switchport mode trunk**

* Configure an interface to be a trunk

// Router Sub-Interface Configuration

Router(config)# **interface [*id*].[*vlan id*]**

* Enter sub-interface configuration mode

Sub-Interfaces are used in conjunction with vlans. To route vlans, a sub-interface must be created for each respective VLAN.

Router(config-subif)# **encapsulation dot1Q [*vlan id*]**

* Configures the interface to use IEEE 802.1Q encapsulation on frames

While 802.1Q is not the only networking standard supporting vlans, it is the most prominent. The vlan id should be a vlan existing on a connected switch desired to be routed.

Router(config-subif)# **ip address [*ip*] [*mask*]**

* Configure an IP for the sub-interface, just like any normal interface

// Dynamic NAT Configuration

Router(config-if)# **ip nat [*inside/outside*]**

* Configure an interface to be internal or external

Inside interfaces are translated through the outside interface.

Router(config-if)# **access-list [*#*] permit [*network address*] [*wildcard mask*]**

* Create an access list that permits a subnet

The subnet specified here should be on an *inside* interface. NAT will translate the subnet out the *outside* interface. Keep note of the [*#*] defined, for that will be used in a later command.

Router(config-if)# **ip nat inside source list [*#*] interface [*id*] overload**

* Enable the translation of an access list through an outside interface

The source list [*#*] should be an access list created earlier; the *interface id* should be the outside interface.

// Wireless LAN Controller Console Commands

(Cisco Controller) > **config interface create [*name*] [*vlan id*]**

* Create an interface with a name and vlan

(Cisco Controller) > **config interface address management [*ip*] [*mask*] [*default gateway*]**

* Configure the pre-defined management interface

There are other pre-defined interfaces, such as the virtual interface, that are edited the same way. Before an interface can be edited, it needs to be disabled.

(Cisco Controller) > **config interface address dynamic-interface [*name*] [*ip*] [*mask*] [*default* *gateway*]**

* Configure a user-defined interface

Interfaces are mapped to wlans. Traffic that the interface receives from the wlan is tagged with interface’s vlan. In other words, each wlan typically is part of a unique vlan. Before an interface can be edited, it needs to be disabled.

(Cisco Controller) > **config wlan create [*#*] [*interface name*] [*ssid*]**

* Create a wlan and map it to an interface

The SSID is the name of the wireless network that clients will join. Each wlan is numbered with no repeats. The model WLC that I use supports up to 512 wlans.

(Cisco Controller) > **config interface vlan [*interface name*] [*vlan id*]**

* Change the vlan of a defined interface

Configurations

Router

Router# show running-config

boot-start-marker

boot system flash bootflash:isr4300-universalk9.16.09.08.SPA.bin

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

ip dhcp excluded-address 10.4.20.129

ip dhcp excluded-address 10.4.20.254

ip dhcp excluded-address 10.4.20.253

ip dhcp excluded-address 10.0.10.1

ip dhcp excluded-address 10.0.10.254

ip dhcp pool webuidhcp

ip dhcp pool USER-WLAN

network 10.0.10.0 255.255.255.0

default-router 10.0.10.1

dns-server 8.8.8.8

ip dhcp pool AP-MANAGEMENT

network 10.4.20.128 255.255.255.128

default-router 10.4.20.1

option 43 hex f104.0a04.14fe

login on-success log

subscriber templating

multilink bundle-name authenticated

crypto pki trustpoint TP-self-signed-2219300048

enrollment selfsigned

subject-name cn=IOS-Self-Signed-Certificate-2219300048

revocation-check none

rsakeypair TP-self-signed-2219300048

license udi pid ISR4321/K9 sn FLM240607T3

no license smart enable

diagnostic bootup level minimal

spanning-tree extend system-id

redundancy

mode none

interface GigabitEthernet0/0/0

ip address dhcp

ip nat outside

no shut

negotiation auto

interface GigabitEthernet0/0/1

no ip address

no shut

negotiation auto

interface GigabitEthernet0/0/1.10

encapsulation dot1Q 10

ip address 10.0.10.1 255.255.255.0

ip nat inside

interface GigabitEthernet0/0/1.420

encapsulation dot1Q 420

ip address 10.4.20.129 255.255.255.128

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

ip forward-protocol nd

ip http server

ip http authentication local

ip http secure-server

ip tftp source-interface GigabitEthernet0

ip nat inside source list 1 interface GigabitEthernet0/0/0 overload

access-list 1 permit 10.0.10.0 0.0.0.255

control-plane

line con 0

transport input none

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

Switch

Switch#show running-config

no service pad

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

hostname Switch

boot-start-marker

boot-end-marker

no aaa new-model

system mtu routing 1500

vtp domain CCNP

vtp mode transparent

authentication mac-move permit

ip subnet-zero

spanning-tree mode pvst

spanning-tree etherchannel guard misconfig

spanning-tree extend system-id

vlan internal allocation policy ascending

vlan 10

name USER-WLAN

vlan 11

name Voice\_Vlan

vlan 20

name Data

vlan 30

name MGT

vlan 40

name MISC

vlan 50

name NATIVE

vlan 420

name MANAGEMENT

interface FastEthernet0/1

switchport access vlan 10

interface FastEthernet0/2

switchport access vlan 10

interface FastEthernet0/3

switchport access vlan 10

interface FastEthernet0/4

switchport access vlan 10

interface FastEthernet0/5

switchport access vlan 10

interface FastEthernet0/6

switchport access vlan 10

interface FastEthernet0/7

switchport access vlan 10

interface FastEthernet0/8

switchport access vlan 10

interface FastEthernet0/9

switchport access vlan 10

interface FastEthernet0/10

switchport access vlan 10

interface FastEthernet0/11

switchport access vlan 10

interface FastEthernet0/12

switchport access vlan 10

interface FastEthernet0/13

switchport access vlan 420

interface FastEthernet0/14

switchport access vlan 420

interface FastEthernet0/15

switchport access vlan 420

interface FastEthernet0/16

switchport access vlan 420

interface FastEthernet0/17

switchport access vlan 420

interface FastEthernet0/18

switchport access vlan 420

interface FastEthernet0/19

switchport access vlan 420

interface FastEthernet0/20

switchport access vlan 420

interface FastEthernet0/21

switchport access vlan 420

interface FastEthernet0/22

switchport access vlan 420

interface FastEthernet0/23

switchport access vlan 420

interface FastEthernet0/24

switchport trunk encapsulation dot1q

switchport mode trunk

interface GigabitEthernet0/1

switchport trunk encapsulation dot1q

switchport mode trunk

interface GigabitEthernet0/2

interface Vlan1

no ip address

interface Vlan10

ip address 10.0.10.2 255.255.255.0

interface Vlan420

ip address 10.4.20.253 255.255.255.128

ip classless

ip http server

ip sla enable reaction-alerts

line con 0

line vty 5 15

end

Wireless LAN Controller

(Cisco Controller) >show run-config commands

802.11a 11nSupport a-mpdu tx scheduler enable

802.11a 11nSupport a-mpdu tx scheduler timeout rt 10

802.11a 11nSupport a-mpdu tx scheduler timeout nrt 200

802.11a 11nSupport a-msdu max-subframes 3

802.11b 11nSupport a-msdu max-subframes 3

802.11a 11nSupport a-msdu max-length 8k

802.11b 11nSupport a-msdu max-length 8k

802.11a 11nSupport mcs tx 8 disable

802.11a 11nSupport mcs tx 9 disable

802.11a beacon range 0

802.11a rx-sop threshold auto default

802.11a cca threshold 0 default

802.11a multicast buffer 0

802.11a multicast data-rate 0 default

802.11a cac video cac-method static

802.11a max-clients 200

802.11a dfs-peakdetect enable

802.11b 11nSupport a-mpdu tx scheduler enable

802.11b 11nSupport a-mpdu tx scheduler timeout rt 10

802.11b 11nSupport a-mpdu tx scheduler timeout nrt 200

802.11b beacon range 0

802.11b rx-sop threshold auto default

802.11b cca threshold 0 default

802.11b multicast buffer 0

802.11b multicast data-rate 0 default

802.11b cac video cac-method static

802.11b max-clients 200

802.11h channelswitch enable loud

aaa auth mgmt local radius

acl url-acl disable

flexconnect fallback-radio-shut disable

advanced 802.11a channel dca interval 0

advanced 802.11a channel dca startup-interval 0

advanced 802.11a channel dca anchor-time 0

advanced 802.11a channel dca chan-width 20

advanced 802.11a channel dca best-width-max 80

advanced 802.11a channel dca sensitivity 15

advanced 802.11a channel dca min-metric -95

advanced 802.11a channel delete 20

advanced 802.11a channel delete 26

advanced 802.11a channel delete 100

advanced 802.11a channel delete 104

advanced 802.11a channel delete 108

advanced 802.11a channel delete 112

advanced 802.11a channel delete 116

advanced 802.11a channel delete 120

advanced 802.11a channel delete 124

advanced 802.11a channel delete 128

advanced 802.11a channel delete 132

advanced 802.11a channel delete 136

advanced 802.11a channel delete 140

advanced 802.11a channel delete 144

advanced 802.11b channel dca interval 0

advanced 802.11b channel dca startup-interval 0

advanced 802.11b channel dca anchor-time 0

advanced 802.11b channel dca sensitivity 10

advanced 802.11b channel dca min-metric -95

location info rogue extended

location rssi-half-life tags 0

location rssi-half-life client 0

location rssi-half-life rogue-aps 0

location expiry tags 5

location expiry client 5

location expiry calibrating-client 5

location expiry rogue-aps 5

advanced 802.11b client-network-preference default

advanced 802.11a client-network-preference default

advanced backup-controller primary

advanced backup-controller secondary

advanced backup-controller

advanced backup-controller

advanced sip-snooping-ports 0 0

advanced eap bcast-key-interval 3600

advanced 802.11-abgn pak-rssi-location threshold -100

advanced 802.11-abgn pak-rssi-location trigger-threshold 10

advanced 802.11-abgn pak-rssi-location reset-threshold 8

advanced 802.11-abgn pak-rssi-location ntp 17.13.78.16

advanced 802.11-abgn pak-rssi-location timeout 3

advanced hotspot cmbk-delay 1

ap cert-expiry-ignore mic enable

ap syslog host global ::

ap dtls-cipher-suite RSA-AES128-SHA

ap dtls-wlc-mic sha2

auth-list add mic 88:5a:92:bd:7d:df

cdp advertise-v2 enable

cts sxp disable

cts sxp connection default password \*\*\*\*

cts sxp retry period 120

cts sxp sxpversion 2

custom-web webmessage “”

database size 2048

dhcp opt-82 remote-id ap-mac

qos qosmap disable

qos qosmap trust-dscp-upstream disable

flexconnect group default-flex-group add

flexconnect group default-flex-group radius ap server-key <hidden>

flexconnect group default-flex-group radius ap authority id 436973636f0000000000000000000000

flexconnect group default-flex-group radius ap authority info Cisco A\_ID

flexconnect group default-flex-group http-proxy ip-address 0.0.0.0 http-proxy port 0

flexconnect group default-flex-group template-vlan-map add none

local-auth method fast server-key \*\*\*\*

interface create user-wlan 10

interface address management 10.4.20.254 255.255.255.128 10.4.20.129

interface address dynamic-interface user-wlan 10.0.10.254 255.255.255.0 10.0.10.1

interface address virtual 192.0.2.1

interface dhcp management primary 10.0.0.1

interface vlan management 420

interface vlan user-wlan 10

interface nasid none user-wlan

nasid apgroup default-group

wlan nasid none 1

wlan nasid none 2

interface port management 13

interface port user-wlan 13

mdns snooping disable

mdns policy service-group create "default-mdns-policy" "Default Access Policy created by WLC"

mdns policy service-group user-role add default-mdns-policy admin

mdns profile create "default-mdns-profile"

mdns service create "AirPrint" \_ipp.\_tcp.local. origin All LSS disable query enable

mdns service create "AirTunes" \_raop.\_tcp.local. origin All LSS disable query enable

mdns service create "AppleTV" \_airplay.\_tcp.local. origin All LSS disable query enable

mdns service create "HP\_Photosmart\_Printer\_1" \_universal.\_sub.\_ipp.\_tcp.local. origin All LSS disable query enable

mdns service create "HP\_Photosmart\_Printer\_2" \_cups.\_sub.\_ipp.\_tcp.local. origin All LSS disable query enable

mdns service create "Printer" \_printer.\_tcp.local. origin All LSS disable query enable

mdns profile service add "default-mdns-profile" "AirPrint"

mdns profile service add "default-mdns-profile" "AirTunes"

mdns profile service add "default-mdns-profile" "AppleTV"

mdns profile service add "default-mdns-profile" "HP\_Photosmart\_Printer\_1"

mdns profile service add "default-mdns-profile" "HP\_Photosmart\_Printer\_2"

mdns profile service add "default-mdns-profile" "Printer"

mdns query interval 15

wlan mdns enable 1

wlan mdns enable 2

wlan mdns profile 1 "default-mdns-profile"

wlan mdns profile 2 "default-mdns-profile"

ipv6 ra-guard ap enable

ipv6 capwap udplite enable all

ipv6 multicast mode unicast

lag enable

load-balancing aggressive enable

load-balancing window 5

wlan apgroup add default-group

wlan apgroup qinq tagging eap-sim-aka default-group enable

wlan apgroup interface-mapping add default-group 1 management

wlan apgroup interface-mapping add default-group 2 user-wlan

wlan apgroup nac-snmp disable default-group 1

wlan apgroup nac-snmp disable default-group 2

memory monitor errors enable

memory monitor leak thresholds 10000 30000

Outdoor Mesh Ext.UNII B Domain channels: Disable

mesh security rad-mac-filter disable

mesh security rad-mac-filter disable

mesh security eap

mesh background-scanning disable

mesh backhaul rrm disable

mesh backhaul rrm auto-rf global

mesh lsc advanced ap-provision open-window enable

mgmtuser add cisco \*\*\*\* read-write

mgmtuser termination-interval 0

mobility dscp 0

network http-proxy ip 0.0.0.0

network http-proxy Port 80

network webmode enable

network secureweb csrfcheck enable

network multicast igmp snooping enable

network multicast mld snooping enable

network profiling http-port 80

network ap-priority disabled

network rf-network-name default

network client-ip-conflict-detection disable

pmipv6 mag binding maximum AP 250

qos protocol-type bronze dot1p

qos protocol-type silver dot1p

qos protocol-type gold dot1p

qos protocol-type platinum dot1p

qos priority bronze background background background

qos priority gold video video video

qos priority platinum voice voice voice

qos priority silver besteffort besteffort besteffort

qos dot1p-tag silver 0

qos dot1p-tag gold 4

qos dot1p-tag platinum 5

radius callStationIdType macaddr

radius auth callStationIdType ap-macaddr-ssid

radius fallback-test mode passive

radius fallback-test username cisco-probe

radius fallback-test interval 300

radius dns disable

radius dns auth network enable

radius dns auth management enable

radius dns acct network enable

radius dns auth rfc3576 disable

tacacs dns disable

rogue detection report-interval 10

rogue detection min-rssi -90

rogue detection transient-rogue-interval 0

rogue detection client-threshold 0

rogue detection security-level custom

rogue ap aaa-auth disable

rogue ap aaa-auth polling-interval 0

rogue ap ssid alarm

rogue ap valid-client alarm

rogue adhoc enable

rogue adhoc alert

rogue ap rldp disable

rogue ap rldp schedule disable

rogue auto-contain level 1

rogue containment flex-connect disable

rogue containment auto-rate disable

rogue client aaa disable

rogue client mse disable

snmp version v2c enable

snmp version v3 enable

snmp snmpEngineId 0000376300005e200101fea9

switchconfig strong-pwd case-check enabled

switchconfig strong-pwd consecutive-check enabled

switchconfig strong-pwd default-check enabled

switchconfig strong-pwd username-check enabled

switchconfig strong-pwd position-check disabled

switchconfig strong-pwd case-digit-check disabled

switchconfig strong-pwd minimum upper-case 0

switchconfig strong-pwd minimum lower-case 0

switchconfig strong-pwd minimum digits-chars 0

switchconfig strong-pwd minimum special-chars 0

switchconfig strong-pwd min-length 3

sysname THEREALCONTROLLER

stats-timer realtime 5

stats-timer normal 180

tacacs fallback-test interval 0

rf-profile create 802.11a High-Client-Density-802.11a

rf-profile create 802.11b High-Client-Density-802.11bg

rf-profile create 802.11a Low-Client-Density-802.11a

rf-profile create 802.11b Low-Client-Density-802.11bg

rf-profile create 802.11a Typical-Client-Density-802.11a

rf-profile create 802.11b Typical-Client-Density-802.11bg

rf-profile tx-power-min 7 High-Client-Density-802.11a

rf-profile tx-power-min 7 High-Client-Density-802.11bg

rf-profile tx-power-control-thresh-v1 -65 High-Client-Density-802.11a

rf-profile tx-power-control-thresh-v1 -60 Low-Client-Density-802.11a

rf-profile tx-power-control-thresh-v1 -65 Low-Client-Density-802.11bg

rf-profile data-rates 802.11a disabled 6 High-Client-Density-802.11a

rf-profile data-rates 802.11a disabled 9 High-Client-Density-802.11a

rf-profile data-rates 802.11a mandatory 12 High-Client-Density-802.11a

rf-profile data-rates 802.11a supported 18 High-Client-Density-802.11a

rf-profile data-rates 802.11a mandatory 24 High-Client-Density-802.11a

rf-profile data-rates 802.11a supported 36 High-Client-Density-802.11a

rf-profile data-rates 802.11a supported 48 High-Client-Density-802.11a

rf-profile data-rates 802.11a supported 54 High-Client-Density-802.11a

rf-profile data-rates 802.11b disabled 1 High-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 2 High-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 5.5 High-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 11 High-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 6 High-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 9 High-Client-Density-802.11bg

rf-profile data-rates 802.11b mandatory 12 High-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 18 High-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 24 High-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 36 High-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 48 High-Client-Density-802.11bg

rf-profile data-rates 802.11a mandatory 6 Low-Client-Density-802.11a

rf-profile data-rates 802.11a supported 9 Low-Client-Density-802.11a

rf-profile data-rates 802.11a mandatory 12 Low-Client-Density-802.11a

rf-profile data-rates 802.11a supported 18 Low-Client-Density-802.11a

rf-profile data-rates 802.11a mandatory 24 Low-Client-Density-802.11a

rf-profile data-rates 802.11a supported 36 Low-Client-Density-802.11a

rf-profile data-rates 802.11a supported 48 Low-Client-Density-802.11a

rf-profile data-rates 802.11a supported 54 Low-Client-Density-802.11a

rf-profile data-rates 802.11b mandatory 1 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b mandatory 2 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b mandatory 5.5 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b mandatory 11 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 6 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 9 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 12 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 18 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 24 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 36 Low-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 48 Low-Client-Density-802.11bg

rf-profile data-rates 802.11a mandatory 6 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a supported 9 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a mandatory 12 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a supported 18 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a mandatory 24 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a supported 36 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a supported 48 Typical-Client-Density-802.11a

rf-profile data-rates 802.11a supported 54 Typical-Client-Density-802.11a

rf-profile data-rates 802.11b disabled 1 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 2 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 5.5 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 11 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b disabled 6 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 9 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b mandatory 12 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 18 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 24 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 36 Typical-Client-Density-802.11bg

rf-profile data-rates 802.11b supported 48 Typical-Client-Density-802.11bg

rf-profile rx-sop threshold medium High-Client-Density-802.11a

rf-profile rx-sop threshold medium High-Client-Density-802.11bg

rf-profile rx-sop threshold low Low-Client-Density-802.11a

rf-profile rx-sop threshold low Low-Client-Density-802.11bg

rf-profile coverage data -90 Low-Client-Density-802.11a

rf-profile coverage data -90 Low-Client-Density-802.11bg

rf-profile coverage voice -90 Low-Client-Density-802.11a

rf-profile coverage voice -90 Low-Client-Density-802.11bg

rf-profile coverage exception 2 Low-Client-Density-802.11a

rf-profile coverage exception 2 Low-Client-Density-802.11bg

rf-profile channel delete 20 High-Client-Density-802.11a

rf-profile channel delete 26 High-Client-Density-802.11a

rf-profile channel delete 100 High-Client-Density-802.11a

rf-profile channel delete 104 High-Client-Density-802.11a

rf-profile channel delete 108 High-Client-Density-802.11a

rf-profile channel delete 112 High-Client-Density-802.11a

rf-profile channel delete 116 High-Client-Density-802.11a

rf-profile channel delete 120 High-Client-Density-802.11a

rf-profile channel delete 124 High-Client-Density-802.11a

rf-profile channel delete 128 High-Client-Density-802.11a

rf-profile channel delete 132 High-Client-Density-802.11a

rf-profile channel delete 136 High-Client-Density-802.11a

rf-profile channel delete 140 High-Client-Density-802.11a

rf-profile channel delete 144 High-Client-Density-802.11a

rf-profile channel delete 20 Low-Client-Density-802.11a

rf-profile channel delete 26 Low-Client-Density-802.11a

rf-profile channel delete 100 Low-Client-Density-802.11a

rf-profile channel delete 104 Low-Client-Density-802.11a

rf-profile channel delete 108 Low-Client-Density-802.11a

rf-profile channel delete 112 Low-Client-Density-802.11a

rf-profile channel delete 116 Low-Client-Density-802.11a

rf-profile channel delete 120 Low-Client-Density-802.11a

rf-profile channel delete 124 Low-Client-Density-802.11a

rf-profile channel delete 128 Low-Client-Density-802.11a

rf-profile channel delete 132 Low-Client-Density-802.11a

rf-profile channel delete 136 Low-Client-Density-802.11a

rf-profile channel delete 140 Low-Client-Density-802.11a

rf-profile channel delete 144 Low-Client-Density-802.11a

rf-profile channel delete 20 Typical-Client-Density-802.11a

rf-profile channel delete 26 Typical-Client-Density-802.11a

rf-profile channel delete 100 Typical-Client-Density-802.11a

rf-profile channel delete 104 Typical-Client-Density-802.11a

rf-profile channel delete 108 Typical-Client-Density-802.11a

rf-profile channel delete 112 Typical-Client-Density-802.11a

rf-profile channel delete 116 Typical-Client-Density-802.11a

rf-profile channel delete 120 Typical-Client-Density-802.11a

rf-profile channel delete 124 Typical-Client-Density-802.11a

rf-profile channel delete 128 Typical-Client-Density-802.11a

rf-profile channel delete 132 Typical-Client-Density-802.11a

rf-profile channel delete 136 Typical-Client-Density-802.11a

rf-profile channel delete 140 Typical-Client-Density-802.11a

rf-profile channel delete 144 Typical-Client-Density-802.11a

rf-profile client-network-preference default High-Client-Density-802.11a

rf-profile client-network-preference default High-Client-Density-802.11bg

rf-profile client-network-preference default Low-Client-Density-802.11a

rf-profile client-network-preference default Low-Client-Density-802.11bg

rf-profile client-network-preference default Typical-Client-Density-802.11a

rf-profile client-network-preference default Typical-Client-Density-802.11bg

trapflags client nac-alert enable

trapflags client webAuthUserLogin enable

trapflags client webAuthUserLogout enable

trapflags ap ssidKeyConflict disable

trapflags ap timeSyncFailure disable

trapflags mfp disable

trapflags adjchannel-rogueap disable

trapflags mesh excessive hop count disable

trapflags mesh sec backhaul change disable

trapflags mesh psk auth failure disable

wlan create 1 Test1 Test

wlan create 2 user-wlan CONNECTTOME

wlan nac snmp disable 1

wlan nac snmp disable 2

wlan nac radius disable 1

wlan nac radius disable 2

wlan interface 2 user-wlan

wlan multicast interface 1 disable

wlan multicast interface 2 disable

wlan band-select allow disable 1

wlan band-select allow disable 2

wlan load-balance allow disable 1

wlan load-balance allow disable 2

wlan assisted-roaming prediction disable 1

wlan assisted-roaming prediction disable 2

wlan assisted-roaming neighbor-list disable 1

wlan assisted-roaming neighbor-list enable 2

wlan assisted-roaming dual-list disable 1

wlan assisted-roaming dual-list disable 2

wlan dms enable 2

wlan bssmaxidle enable 1

wlan bssmaxidle enable 2

wlan bss-transition enable 2

wlan bss-transition disassociation-imminent timer 200 1

wlan bss-transition disassociation-imminent timer 200 2

wlan bss-transition disassociation-imminent oproam-timer 40 1

wlan bss-transition disassociation-imminent oproam-timer 40 2

wlan multicast buffer disable 0 1

wlan multicast buffer disable 0 2

wlan session-timeout 1 1800

wlan session-timeout 2 1800

wlan flexconnect local-switching 1 disable

wlan flexconnect local-switching 2 disable

wlan flexconnect learn-ipaddr 1 enable

wlan flexconnect learn-ipaddr 2 enable

wlan wgb broadcast-tagging disable 1

wlan wgb broadcast-tagging disable 2

wlan security wpa disable 2

wlan security splash-page-web-redir disable 1

wlan security splash-page-web-redir disable 2

wlan security wpa akm 802.1x enable 1

wlan security wpa akm cckm timestamp-tolerance 1000 1

wlan security wpa akm cckm timestamp-tolerance 1000 2

wlan security ft adaptive enable 2

wlan security wpa gtk-random disable 1

wlan security wpa gtk-random disable 2

wlan security pmf association-comeback 1 1

wlan security pmf association-comeback 1 2

wlan security pmf saquery-retrytimeout 200 1

wlan security pmf saquery-retrytimeout 200 2

wlan dhcp\_server 2 10.0.10.1 required

wlan profiling radius dhcp disable 1

wlan profiling radius http disable 1

wlan profiling radius dhcp disable 2

wlan profiling radius http disable 2

wlan enable 1

wlan enable 2

license boot base

WMM-AC disabled

HS2 QOS disabled

coredump disable

media-stream multicast-direct disable

media-stream message url

media-stream message email

media-stream message phone

media-stream message note denial

media-stream message state disable

802.11a media-stream multicast-direct enable

802.11b media-stream multicast-direct enable

802.11a media-stream multicast-direct radio-maximum 0

802.11b media-stream multicast-direct radio-maximum 0

802.11a media-stream multicast-direct client-maximum 0

802.11b media-stream multicast-direct client-maximum 0

802.11a media-stream multicast-direct admission-besteffort disable

802.11b media-stream multicast-direct admission-besteffort disable

802.11a media-stream video-redirect enable

802.11b media-stream video-redirect enable

ipv6 neighbor-binding timers reachable-lifetime 300

ipv6 neighbor-binding timers stale-lifetime 86400

ipv6 neighbor-binding timers down-lifetime 30

ipv6 neighbor-binding ra-throttle disable

ipv6 neighbor-binding ra-throttle allow at-least 1 at-most 1

ipv6 neighbor-binding ra-throttle max-through 10

ipv6 neighbor-binding ra-throttle throttle-period 600

ipv6 neighbor-binding ra-throttle interval-option passthrough

ipv6 ns-mcast-fwd disable

ipv6 na-mcast-fwd enable

ipv6 enable

nmheartbeat disable

ipv6 slaac service-port disable

sys-nas

tunnel eogre heart-beat interval 60

tunnel eogre heart-beat primary-fallback-timeout 30

tunnel eogre heart-beat max-skip-count 3

wlan mobility selective-reanchoring disable 1

wlan mobility selective-reanchoring disable 2

cloud-services wsa mode Disable

network assurance server url

WSA Backhaul SSID

WSA Backhaul Username

WSA Backhaul Authentication Type psk

Eap Type ........................................ <none>

cloud-services server url

cloud-services cmx disabled

Conclusion

While it may have taken some time familiarizing myself with the features and configurations of a Wireless LAN Controller, I will say the payoff is worth the effort when managing many access points. With only two access points in my arsenal, it would likely have been better to configure them standalone in a production environment. However, this was not for a production environment, as I was configuring the WLC for educational purposes.