Introduction to Socket Programming

Md. Tareq Mahmood

Network

client
client
client
client
Internet
ISP router
(DHCP enabled)
WLAN router
WLAN router

IP Address

- Identification of a machine in a network.
- ‘ipconfig’ or ‘ifconfig’ command to see ip address.

IP Address

A machine can have multiple IP addresses for several reasons, depending on its configuration and role in a network. Here are some common reasons:

1. Multiple Network Interfaces (NICs)
A machine with multiple network interface cards (NICs) (e.g., Ethernet, Wi-Fi) will have a unique IP address for each interface.
Example: A server with one NIC for internal communication (192.168.1.10) and another for external access (203.0.113.5).

2. Virtual Network Interfaces
Virtual machines (VMs) and Docker containers running on a host often get their own IP addresses.
VPN connections may create virtual interfaces with separate IPs.

3. IPv4 and IPv6 Coexistence
A machine may have both an IPv4 address (e.g., 192.168.1.10) and an IPv6 address (e.g., 2001:db8::1) for compatibility.

A machine may have multiple network.
For each network, it will have a different IP address.

Network Interface

- Number of active network interface = number of ip address in that device
- Different IP Address for each Interface

Port

Why 2^16 (65,536) Ports?
A port number is a 16-bit value (ranging from 0 to 65,535).
This allows multiple applications to run on the same machine and use different ports for communication.

- Endpoint/channel for communication for different programs
- 2^16 ports, some are reserved
- A computer process must acquire a port for network communication
- A logical construct
- ‘netstat’ command to see ports in use

Reserved & Well-Known Ports
Some ports are reserved for specific protocols and services:

Well-Known Ports (0–1023) ---> Reserved for system services.
22 ---> SSH
25 ---> SMTP
53 ---> DNS
80 --> HTTP
443 --> HTTPS

Registered Ports (1024–49,151)--->Used by user applications.
3306 ---> MySQL
5432 --->PostgreSQL
25565--->Minecraft Server

Connection Establishment

- You need (IP address, Port) to establish a connection to remote PC
- A program must be running to that PC to accept your connection
- Some program must be running on that port

- Example: buet.ac.bd:443

Why does buet.ac.bd:443 work?
443 is the default HTTPS (secure web) port.
Web servers (like Apache, Nginx) typically listen on port 443 for secure connections.
When you visit https://buet.ac.bd, your browser automatically connects to buet.ac.bd:443.

- Error for, buet.ac.bd:120

Why does buet.ac.bd:120 give an error?
Port 120 is not commonly used for web services.
The BUET server is likely not configured to listen on port 120.
Firewall rules may block requests to non-standard ports.
If the server does not have any application running on port 120, the connection will be refused or timed out.

How to check which ports are open?
You can use the telnet command to test:

telnet buet.ac.bd 120

Socket

- Represents a single connection between two network applications
- Number of connection = number of sockets
- A socket must have these informations to communicate
  - Remote IP
  - Remote Port
  - Local Port
- A socket need these bufferes to operate
  - Input buffer
  - Output buffer

many : one

Socket vs Port

Example: A Web Server on Port 80
A web server (e.g., Apache) binds to 0.0.0.0:80, meaning it listens on all interfaces at port 80.
When multiple clients connect, the server creates unique socket connections for each client.
Client 1: 192.168.1.5:5000 --> 203.0.113.1:80
Client 2: 192.168.1.6:5001 --> 203.0.113.1:80
Client 3: 192.168.1.7:5002 --> 203.0.113.1:80
Although they all use port 80 on the server, the client ports differ, creating unique sockets.

- Multiple sockets can be using same ports
- But a port must be acquired by only one program

Feature	Socket	Port
Definition	A combination of an IP address and a port number, forming an endpoint for network communication.	A numerical identifier assigned to a process or service to handle network traffic.
Structure	Consists of an IP address and a port number (e.g., 192.168.1.1:8080).	Only a number ranging from 0 to 65535.
Purpose	Represents a unique communication channel between two systems.	Identifies a specific process or service on a machine.
Scope	Exists at both ends of a communication link (client and server).	Exists within a system to route network traffic to the correct application.
Uniqueness	A socket is unique across a network (e.g., 192.168.1.1:8080 vs. 192.168.1.1:9090).	A port alone is not unique; it must be paired with an IP to form a socket.
Example	192.168.1.1:8080 (Socket = IP + Port)	8080, 443, 22 (Port only)

A port is a numerical identifier (0–65535) that helps direct network traffic to a specific application or service running on a system. The statement means that a specific port on a machine can be used by only one process at a time for communication.
Why Can Only One Program Use a Port at a Time?
Port Binding:
When a program (e.g., a web server) starts, it binds to a port (e.g., 80 for HTTP).
This means the OS ensures that no other program can use that port while it’s occupied.
Avoiding Conflicts:
If two programs try to listen on the same port, they would receive mixed or corrupted data, causing unpredictable behavior.
The OS prevents this by restricting each port to a single listening program.
Error Handling:
If a second program tries to bind to an already occupied port, it gets an "Address already in use" error.

https://stackoverflow.com/questions/152457/what-is-the-difference-between-a-port-and-a-socket

Connect Two PC

1. A listens for connection in a port (6666) using a Server Socket

Server

Client

Connect Two PC

2. B tries connect to A using (192.168.0.101, 6666)

Connect Two PC

3. A accepts B’s connection

Connect Two PC

4. Both of them has a socket of their own

Connect Two PC

5. Both of them knows each other's IP address and Port

Connect Two PC

6. Server keeps listening on same port for new connection

Connect Two PC

7. Multiple client can connect

Different socket, but on same port

TCP

- This whole thing is done using Transmission Control Protocol

15-213

"The course that gives CMU its Zip!"

Web Services
Nov 26, 2002

Topics

- **HTTP**

Web Servers

HTTP (Hypertext Transfer Protocol) is a request-response protocol used for communication between web browsers and web servers. It primarily uses TCP (Transmission Control Protocol) instead of UDP because of reliability, ordered delivery, and connection-oriented communication.

Clients and servers communicate using the HyperText Transfer Protocol (HTTP)

- Client and server establish TCP connection
- Client requests content
- Server responds with requested content
- Client and server close connection (usually)

Current version is HTTP/1.1

- RFC 2616, June, 1999.

The current version of the Hypertext Transfer Protocol (HTTP) is HTTP/3, which was standardized by the Internet Engineering Task Force (IETF) in June 2022.
As of September 2024, HTTP/3 is supported by over 95% of major web browsers and is utilized by approximately 34% of the top 10 million websites

Web Content

Web servers return *content* to clients

- *content:* a sequence of bytes with an associated MIME (Multipurpose Internet Mail Extensions) type

Example MIME types

- `text/html` HTML document
- `text/plain` Unformatted text
- `application/postscript` Postcript document
- `image/gif` Binary image encoded in GIF format
- `image/jpeg` Binary image encoded in JPEG format

JPEG: Joint Photographic Experts Group
GIF: Graphics Interchange Format

Static and Dynamic Content

The content returned in HTTP responses can be either *static* or *dynamic*.

- **Static content: content stored in files and retrieved in response to an HTTP request**
  - **Examples: HTML files, images, audio clips.**
- **Dynamic content: content produced on-the-fly in response to an HTTP request**
  - **Example: content produced by a program executed by the server on behalf of the client.**

Bottom line: *All Web content is associated with a file that is managed by the server.*

URLs

Each file managed by a server has a unique name called a URL (Universal Resource Locator)

URLs for static content:

- `http://www.cs.cmu.edu:80/index.html`
- `http://www.cs.cmu.edu/index.html`
- `http://www.cs.cmu.edu`
  - Identifies a file called `index.html`, managed by a Web server at `www.cs.cmu.edu` that is listening on port 80.

URLs for dynamic content:

- `http://www.cs.cmu.edu:8000/cgi-bin/adder?15000&213`
  - Identifies an executable file called `adder`, managed by a Web server at `www.cs.cmu.edu` that is listening on port 8000, that should be called with two argument strings: `15000` and `213`.

How Clients and Servers Use URLs

Example URL: `http://www.aol.com:80/index.html`

Clients use *prefix* (`http://www.aol.com:80`) **to infer:**

- **What kind of server to contact** (Web server)
- **Where the server is** (`www.aol.com`)
- **What port** it is listening on (80)

Servers use *suffix* (`/index.html`) **to:**

- **Determine if request is for static or dynamic content.**
  - **No hard and fast rules for this.**
  - **Convention: executables reside in `cgi-bin` directory**
- **Find file on file system.**
  - **Initial "/" in suffix denotes home directory for requested content.**
  - **Minimal suffix is "/", which all servers expand to some default home page** (e.g., `index.html`).

Anatomy of an HTTP Transaction

```
unix> telnet www.aol.com 80
Trying 205.188.146.23...
Connected to aol.com.
Escape character is '^]'.
GET / HTTP/1.1
host: www.aol.com

HTTP/1.0 200 OK
MIME-Version: 1.0
Date: Mon, 08 Jan 2001 04:59:42 GMT
Server: NaviServer/2.0 AOLserver/2.3.3
Content-Type: text/html
Content-Length: 42092

<html>
...
</html>
Connection closed by foreign host.
unix>
```

Client: open connection to server
Telnet prints 3 lines to the terminal

Client: request line
Client: required HTTP/1.1 HOST header
Client: empty line terminates headers .
Server: response line
Server: followed by five response headers

Server: expect HTML in the response body
Server: expect 42,092 bytes in the resp body
Server: empty line ("\r\n") terminates hdrs
Server: first HTML line in response body
Server: 766 lines of HTML not shown.
Server: last HTML line in response body
Server: closes connection
Client: closes connection and terminates

HTTP Requests

HTTP request is a *request line*, followed by zero or more *request headers*

Request line: `<method>` `<uri>` `<version>`

- `<version>` is HTTP version of request (`HTTP/1.0` or `HTTP/1.1`)
- `<uri>` is typically URL for proxies, URL suffix for servers.
- `<method>` is either `GET`, `POST`, `OPTIONS`, `HEAD`, `PUT`, `DELETE`, or `TRACE`.

HTTP Requests (cont)

HTTP methods:

- `GET`: Retrieve static or dynamic content
  - Arguments for dynamic content are in URI
  - Workhorse method (99% of requests)
- `POST`: Retrieve dynamic content
  - Arguments for dynamic content are in the request body
- `OPTIONS`: Get server or file attributes
- `HEAD`: Like `GET` but no data in response body
- `PUT`: Write a file to the server!
- `DELETE`: Delete a file on the server!
- `TRACE`: Echo request in response body
  - Useful for debugging.

POST: Retrieve (or Submit) Dynamic Content
How POST retrieves dynamic content?
Unlike GET, POST does not send parameters in the URL.
Instead, parameters are sent in the request body.

Workhorse Method (99% of requests)
GET is the most commonly used HTTP method because:
Browsing websites mostly involves retrieving content.
Search engines, APIs, and most web applications rely on GET for fetching data.

The HEAD method is similar to GET, but the key difference is:
It does not return a response body—only headers.

Why Use HEAD?
Check if a resource exists without downloading it. Retrieve metadata (content type, length, last modified date, etc.).
Improve performance by avoiding unnecessary data transfer.
Check cache validation using ETag or Last-Modified headers.

The TRACE method in HTTP is used for diagnostic purposes. When a client sends an HTTP TRACE request, the server echoes back the received request line, headers, and sometimes the body (if present). This helps in debugging by showing how the request is processed by intermediaries (proxies, firewalls, etc.).

HTTP Requests (cont)

Request headers: `<header name>: <header data>`

- **Provide additional information to the server.**

Major differences between HTTP/1.1 and HTTP/1.0

1
- **HTTP/1.0 uses a new connection for each transaction.**
- **HTTP/1.1 also supports *persistent connections***
  - **multiple transactions over the same connection**
  - `Connection: Keep-Alive`

3
- **HTTP/1.1 requires `HOST` header**
  - `Host: kittyhawk.cmcl.cs.cmu.edu`

2
- **HTTP/1.1 adds additional support for caching**

HTTP/1.0 does not require the Host header because it was designed before virtual hosting became common. In HTTP/1.0:
Each request is meant for a single server (IP address), so there was no need to specify the host.
URLs in the request typically contained only the path (e.g., GET /index.html HTTP/1.0).

HTTP Responses

400 Bad Request --> The server cannot process the request due to client error.
401 Unauthorized --> Authentication required but missing or invalid.
500 Internal Server Error --> A generic server error (unexpected condition)

HTTP response is a *response line* followed by zero or more *response headers*.

Response line:

`<version> <status code> <status msg>`

- **<version> is HTTP version of the response.**
- **<status code> is numeric status.**
- **<status msg> is corresponding English text.**
  - **200 OK Request was handled without error** 100 ---> Continue
  - **403 Forbidden Server lacks permission to access file**
  - **404 Not found Server couldn't find the file.**

Response headers: `<header name>: <header data>`

- **Provide additional information about response**
- `Content-Type:` **MIME type of content in response body.**
- `Content-Length:` **Length of content in response body.**

GET Request to Apache Server From IE Browser

```
GET /test.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
Host: euro.ecom.cmu.edu
Connection: Keep-Alive
CRLF (\r\n)
```

GET Response From Apache Server

```
HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 04:02:15 GMT
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Last-Modified: Thu, 22 Jul 1999 03:33:21 GMT
ETag: "48bb2-4f-37969101"
Accept-Ranges: bytes
Content-Length: 79
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
CRLF
<html>
<head><title>Test page</title></head>
<body>
<h1>Test page</h1>
</html>
```

Proxies

A *proxy* is an intermediary between a client and an *origin server*.

- To the client, the proxy acts like a server.
- To the server, the proxy acts like a client.

Why Proxies?

Can perform useful functions as requests and responses pass by

- **Examples: Caching, logging, anonymization**

Anonymity --> The server doesn't see the actual client IP.
Content Filtering ---> The proxy can modify or block certain requests.
Caching ---> The proxy can cache responses to improve speed.

Basic Switch and End Device Configuration

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Cisco IOS Access

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Operating Systems

Cisco Internetwork Operating System (IOS) can be accessed using different methods, depending on the device type and network setup.

- **Shell** - The user interface that allows users to request specific tasks from the computer. These requests can be made either through the CLI or GUI interfaces.
- **Kernel** - Communicates between the hardware and software of a computer and manages how hardware resources are used to meet software requirements.
- **Hardware** - The physical part of a computer including underlying electronics.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Cisco IOS Access

Access Methods

For secure remote management --> Use SSH
For local access (no network needed) --> Use Console
For legacy remote access -->Use Telnet (not recommended)
For dial-in access ---> Use AUX
For web-based management ---> Use HTTP/HTTPS

- **Console** – A physical management port used to access a device in order to provide maintenance, such as performing the initial configurations.
- **Secure Shell (SSH)** – Establishes a secure remote CLI connection to a device, through a virtual interface, over a network. (Note: This is the recommended method for remotely connecting to a device.)
- **Telnet** – Establishes an insecure remote CLI connection to a device over the network. (Note: User authentication, passwords and commands are sent over the network in plaintext.)

A CLI (Command-Line Interface) connection refers to accessing and managing a device (like a router, switch, or server) using text-based commands instead of a graphical interface. In networking, CLI is commonly used to configure and troubleshoot devices like Cisco routers and switches.

Terminal Emulation Programs

- Terminal emulation programs are used to connect to a network device by either a console port or by an SSH/Telnet connection.
- There are several terminal emulation programs to chose from such as PuTTY, Tera Term, etc

and/or its affiliates. All rights reserved. Cisco Confidential

IOS Navigation

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Primary Command Modes

User EXEC Mode:

- Allows access to only a limited number of basic monitoring commands
- Identified by the CLI prompt that ends with the > symbol

```
Router>

Switch>
```

Privileged EXEC Mode:

- Allows access to all commands and features
- Identified by the CLI prompt that ends with the # symbol

```
Router#

Switch#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Navigation Between IOS Modes

- **Privileged EXEC Mode:**
  - To move from user EXEC mode to privilege EXEC mode, use the **enabled** command.

```
Switch> enable
Switch#
```

- **Global Configuration Mode:**
  - To move in and out of global configuration mode, use the **configure terminal** command. To return to privilege EXEC mode, use the **exit** command.

```
Switch(config)#
Switch(config)#exit
Switch#
```

- **Line Configuration Mode:**
  - To move in and out of line configuration mode, use the **line** command followed by the management line type. To return to global configuration mode, use the **exit** command.

```
Switch(config)#line console 0
Switch(config-line)#exit
Switch(config)#
```

Line Configuration Mode is a command-line interface (CLI) mode in Cisco networking devices (like routers and switches) that allows you to configure terminal lines. These lines include console, auxiliary (AUX), and virtual terminal (VTY) lines, which control remote and local access to the device.

cisco

The Command Structure

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Basic IOS Command Structure

- **Keyword** – This is a specific parameter defined in the operating system (in the figure, **ip protocols**).
- **Argument** - This is not predefined; it is a value or variable defined by the user (in the figure, **192.168.10.5**).

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Hot Keys and Shortcuts

- The IOS CLI provides hot keys and shortcuts that make configuring, monitoring, and troubleshooting easier.
- Commands and keywords can be shortened to the minimum number of characters that identify a unique selection. For example, the **configure** command can be shortened to **conf** because **configure** is the only command that begins with **conf**.

```
Router#con
% Ambiguous command: "con"
Router#con?
configure  connect
```

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Hot Keys and Shortcuts

- When a command output produces more text than can be displayed in a terminal window, the IOS will display a "**--More--**" prompt. The table below describes the keystrokes that can be used when this prompt is displayed.

Keystroke	Description
Enter Key	Displays the next line.
Space Bar	Displays the next screen.
Any other key	Ends the display string, returning to privileged EXEC mode.

- The table below lists commands that can be used to exit out of an operation.

Keystroke	Description
Ctrl-C	When in any configuration mode, ends the configuration mode and returns to privileged EXEC mode.
//	
Ctrl-Z	When in any configuration mode, ends the configuration mode and returns to privileged EXEC mode.
Ctrl-Shift-6	All-purpose break sequence used to abort DNS lookups, traceroutes, pings, etc.

The Ctrl-Shift-6 key combination is a special break sequence used in Cisco IOS devices (routers and switches). It allows you to interrupt or abort certain ongoing processes in the command-line interface (CLI), such as:
DNS lookups,Traceroutes,Pings,Telnet or SSH sessions,Other long-running commands

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Basic Device Configuration

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Device Names

- The first configuration command on any device : unique hostname.
- By default, all devices are assigned a factory default name. For example, a Cisco IOS switch is "Switch."

```
Switch# configure terminal
Switch(config)# hostname Sw-Floor-1
Sw-Floor-1(config)#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Passwords

line ---> This command is used to configure a specific line (console, VTY, or auxiliary).
console ---> Specifies that you are configuring the console line, which is the physical port for direct device access.
0 ---> The console port is always line 0 because a Cisco device has only one console port.

Securing user EXEC mode access:

- First enter line console configuration mode using the **line console 0** command in global configuration mode.
- Next, specify the user EXEC mode password using the **password** *password* command.
- Finally, enable user EXEC access using the **login** command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# line console 0
Sw-Floor-1(config-line)# password cisco
Sw-Floor-1(config-line)# login
Sw-Floor-1(config-line)# end
Sw-Floor-1#
```

Securing privileged EXEC mode access:

- First enter global configuration mode.
- Next, use the **enable secret** *password* command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# enable secret class
Sw-Floor-1(config)# exit
Sw-Floor-1#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Passwords: VTY

VTY (Virtual Teletype) lines are used for remote access to a Cisco device via Telnet or SSH. To secure remote access, you must set a password and enable authentication.

Securing VTY line access:

- First enter line VTY configuration mode using the **line vty 0 15** command in global configuration mode.
- Next, specify the VTY password using the **password** *password* command.
- Finally, enable VTY access using the **login** command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# line vty 0 15
Sw-Floor-1(config-line)# password cisco
Sw-Floor-1(config-line)# login
Sw-Floor-1(config-line)# end
Sw-Floor-1#
```

Selects VTY lines 0–15 for configuration

- Note: VTY lines enable remote access using Telnet or SSH to the device. Many Cisco switches support up to 16 VTY lines that are numbered 0 to 15.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Encrypt Passwords

- The startup-config and running-config files display most passwords in plaintext.
- To encrypt all plaintext passwords, use the **service password-encryption** global config command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# service password-encryption
Sw-Floor-1(config)# exit
Sw-Floor-1#
```

- Use the **show running-config** command to verify that the passwords on the device are now encrypted.

```
Sw-Floor-1# show running-config
!

!
line con 0
password 7 094F471A1A0A
login
!
Line vty 0 4
Password 7 03095A0F034F38435B49150A1819
Login
!
!
end
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Banner Messages

- A banner message is important to warn unauthorized personnel from attempting to access the device.
- To create a banner message of the day on a network device, use the **banner motd #** *the message of the day* **#** global config command.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# banner motd #Authorized Access Only!#
```

The banner will be displayed on attempts to access the device.

Banner Type	Purpose	When It Appears
MOTD (Message of the Day)	General message for all users	Before login
Login	Legal notice or security message	Before password prompt
Exec	Message after login	After successful login

```
Press RETURN to get started.



Authorized Access Only!

User Access Verification

Password:
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Save Configurations

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configuration Files

- There are two system files that store the device configuration:
  - **startup-config** - This is the saved configuration file that is stored in NVRAM. It contains all the commands that will be used by the device upon startup or reboot. Flash does not lose its contents when the device is powered off.
  - **running-config** - This is stored in Random Access Memory (RAM). It reflects the current configuration. Modifying a running configuration affects the operation of a Cisco device immediately. RAM is volatile memory. It loses all of its content when the device is powered off or restarted.
  - To save changes made to the running configuration to the startup configuration file, use:
  - **copy running-config startup-config**

```
Router#show startup-config
Using 624 bytes
!
version 15.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
```

```
Router#show running-config
Building configuration...

Current configuration : 624 bytes
!
version 15.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Alter the Running Configurations

If changes made to the running config do not have the desired effect and the running-config has not yet been saved, you can restore the device to its previous configuration. To do this you can:

- Remove the changed commands individually.
- Reload the device using the **reload** command in privilege EXEC mode. *Note: This will cause the device to briefly go offline, leading to network downtime.*

```
Router# reload
Proceed with reload? [confirm]
Initializing Hardware ...
```

If the undesired changes were saved to the startup-config, it may be necessary to clear all the configurations using the **erase startup-config** command in privilege EXEC mode.

- After erasing the startup-config, reload the device to clear the running-config file from RAM.

```
Router# erase startup-config
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]
[OK]
Erase of nvram: complete
%SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
Router#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Ports and Addresses

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

IP Addresses

- The use of IP addresses is the primary means of enabling devices to locate one another and establish end-to-end communication on the internet.
- The structure of an IPv4 address is called dotted decimal notation and is represented by four decimal numbers between 0 and 255.
- An IPv4 subnet mask is a 32-bit value that differentiates the network portion of the address from the host portion. Coupled with the IPv4 address, the subnet mask determines to which subnet the device is a member.
- The default gateway address is the IP address of the router that the host will use to access remote networks, including the internet.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

IPv6 Addresses

- IPv6 addresses are 128 bits in length and written as a string of hexadecimal values. Every four bits is represented by a single hexadecimal digit; for a total of 32 hexadecimal values. Groups of four hexadecimal digits are separated by a colon ":".
- IPv6 addresses are not case-sensitive and can be written in either lowercase or uppercase.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure IP Addressing

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Manual IP Address Configuration for End Devices

- End devices on the network need an IP address in order to communicate with other devices on the network.
- IPv4 address information can be entered into end devices manually, or automatically using Dynamic Host Configuration Protocol (DHCP).
  - To manually configure an IPv4 address on a Windows PC, open the **Control Panel > Network Sharing Center > Change adapter settings** and choose the adapter. Next right-click and select **Properties** to display the **Local Area Connection Properties**.
  - Next, click **Properties** to open the **Internet Protocol Version 4 (TCP/IPv4) Properties** window. Then configure the IPv4 address and subnet mask information, and default gateway.

Note: IPv6 addressing and configuration options are similar to IPv4.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Automatic IP Address Configuration for End Devices

- DHCP enables automatic IPv4 address configuration for every end device that is DHCP-enabled.
- End devices are typically by default using DHCP for automatic IPv4 address configuration.
  - To configure DHCP on a Windows PC, open the **Control Panel > Network Sharing Center > Change adapter settings** and choose the adapter. Next right-click and select **Properties** to display the **Local Area Connection Properties**.
  - Next, click **Properties** to open the **Internet Protocol Version 4 (TCP/IPv4) Properties** window, then select **Obtain an IP address automatically** and **Obtain DNS server address automatically**.

Note: IPv6 uses DHCPv6 and SLAAC (Stateless Address Autoconfiguration) for dynamic address allocation.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Switch Virtual Interface Configuration

To access the switch remotely, an IP address and a subnet mask must be configured on the SVI.

To configure an SVI on a switch:

- Enter the **interface vlan 1** command in global configuration mode.
- Next assign an IPv4 address using the **ip address** *ip-address subnet-mask command.*
- Finally, enable the virtual interface using the **no shutdown** command.

```
Switch# configure terminal
Switch(config)# interface vlan 1
Switch(config-if)# ip address 192.168.1.20 255.255.255.0
Switch(config-if)# no shutdown
```

A Switch Virtual Interface (SVI) is a virtual interface on a Layer 3 switch that allows it to route traffic between VLANs. Unlike physical interfaces, SVIs do not require a physical port but act as the Layer 3 gateway for VLANs.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Router

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Basic Router Configuration Steps

- Configure the device name.

```
Router(config)# hostname hostname
```

- Secure privileged EXEC mode.

```
Router(config)# enable secret password
```

- Secure user EXEC mode.

```
Router(config)# line console 0
Router(config-line)# password password
Router(config-line)# login
```

- Secure remote Telnet / SSH access.

```
Router(config)# line vty 0 4
Router(config-line)# password password
Router(config-line)# login
Router(config-line)# transport input {ssh | telnet}
```

- Encrypt all plaintext passwords.

```
Router(config)# service password encryption
```

- Provide legal notification and save the configuration.

```
Router(config)# banner motd # message #
Router(config)# end
Router# copy running-config startup-config
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Basic Router Configuration Example

- Commands for basic router configuration on R1.
- Configuration is saved to NVRAM.

```
R1(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# transport input ssh telnet
R1(config-line)# exit
R1(config)# service password encryption
R1(config)# banner motd #
Enter TEXT message. End with a new line and the #
**********************************************
WARNING: Unauthorized access is prohibited!
**********************************************
R1(config)# exit
R1# copy running-config startup-config
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Interfaces

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Router Interfaces

Configuring a router interface includes issuing the following commands:

```
Router(config)# interface type-and-number
Router(config-if)# description description-text
Router(config-if)# ip address ipv4-address subnet-mask
Router(config-if)# ipv6 address ipv6-address/prefix-length
Router(config-if)# no shutdown
```

- It is a good practice to use the **description** command to add information about the network connected to the interface.
- The **no shutdown** command activates the interface.

Status (Line Status)	Protocol Status	Meaning	Possible Causes
Up	Up	Interface is fully operational and passing traffic.	Normal condition
Up	Down	Physical link is fine, but Layer 2/3 issues exist.	No IP configured - Encapsulation mismatch (PPP/HDLC) - Keepalive issues
Down	Down	Physical link is not active.	Cable unplugged/damaged - Device on other end is powered off
-			Admin disabled (shutdown command)

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Router Interfaces Example

The commands to configure interface G0/0/0 on R1 are shown here:

G --->Stands for Gigabit Ethernet, meaning it supports speeds up to 1 Gbps.
0/0 ---> Indicates the slot and sub-slot in modular switches or routers.
1 --> The interface number within that slot.

```
R1(config)# interface gigabitEthernet 0/0/0
R1(config-if)# description Link to LAN
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:10::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
*Aug  1 01:43:53.435: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0, changed state to down
*Aug  1 01:43:56.447: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0, changed state to up
*Aug  1 01:43:57.447: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0,
changed state to up
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Router Interfaces Example (Cont.)

The commands to configure interface G0/0/1 on R1 are shown here:

```
R1(config)# interface gigabitEthernet 0/0/1
R1(config-if)# description Link to R2
R1(config-if)# ip address 209.165.200.225 255.255.255.252
R1(config-if)# ipv6 address 2001:db8:feed:224::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
*Aug  1 01:46:29.170: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to down
*Aug  1 01:46:32.171: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to up
*Aug  1 01:46:33.171: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/1,
changed state to up
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Verify Interface Configuration

To verify interface configuration use the **show ip interface brief** and **show ipv6 interface brief** commands shown here:

```
R1#  show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   192.168.10.1    YES manual up                    up
GigabitEthernet0/0/1   209.165.200.225 YES manual up                    up
Vlan1                  unassigned      YES unset  administratively down down
```

```
R1#  show ipv6 interface brief
GigabitEthernet0/0/0       [up/up]
    FE80::201:C9FF:FE89:4501
    2001:DB8:ACAD:10::1
GigabitEthernet0/0/1       [up/up]
    FE80::201:C9FF:FE89:4502
    2001:DB8:FEED:224::1
Vlan1                      [administratively down/down]
    unassigned
R1#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Verification Commands

The table summarizes show commands used to verify interface configuration.

Commands	Description
`show ip interface brief` `show ipv6 interface brief`	Displays all interfaces, their IP addresses, and their current status.
`show ip route` `show ipv6 route`	Displays the contents of the IP routing tables stored in RAM.
`show interfaces`	Displays statistics for all interfaces on the device. Only displays the IPv4 addressing information.
`show ip interfaces`	Displays the IPv4 statistics for all interfaces on a router.
`show ipv6 interfaces`	Displays the IPv6 statistics for all interfaces on a router.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Verification Commands (Cont.)

View status of all interfaces with the **show ip interface brief** and **show ipv6 interface brief** commands, shown here:

```
R1# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   192.168.10.1    YES manual up                    up
GigabitEthernet0/0/1   209.165.200.225 YES manual up                    up
Vlan1                  unassigned      YES unset  administratively down down
R1#
```

```
R1# show ipv6 interface brief
GigabitEthernet0/0/0       [up/up]
    FE80::201:C9FF:FE89:4501
    2001:DB8:ACAD:10::1
GigabitEthernet0/0/1       [up/up]
    FE80::201:C9FF:FE89:4502
    2001:DB8:FEED:224::1
Vlan1                      [administratively down/down]
    unassigned
R1#
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Verification Commands (Cont.)

Display the contents of the IP routing tables with the **show ip route** and **show ipv6 route** commands as shown here:

```
R1# show ip route
< output omitted>
Gateway of last resort is not set
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.10.0/24 is directly connected, GigabitEthernet0/0/0
L        192.168.10.1/32 is directly connected, GigabitEthernet0/0/0
      209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.200.224/30 is directly connected, GigabitEthernet0/0/1
L        209.165.200.225/32 is directly connected, GigabitEthernet0/0/1
R1#
```

```
R1# show ipv6 route
<output omitted>
C   2001:DB8:ACAD:10::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   2001:DB8:ACAD:10::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   2001:DB8:FEED:224::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   2001:DB8:FEED:224::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
L   FF00::/8 [0/0]
     via Null0, receive
R1#
```

Configure Verification Commands (Cont.)

Display IPv4 statistics for router interfaces with the **show ip interface** command, as shown here:

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing Common access list is not set
  Outgoing access list is not set
  Inbound Common access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP Flow switching is disabled

<output omitted>

R1#
```

Topics Learned

- The tasks that should be completed when configuring initial settings on a router.
  - Configure the device name.
  - Secure privileged EXEC mode.
  - Secure user EXEC mode.
  - Secure remote Telnet / SSH access.
  - Secure all passwords in the config file.
  - Provide legal notification.
  - Save the configuration.
- For routers to be reachable, the router interfaces must be configured.
  - Using the **no shutdown** command activates the interface. The interface must also be connected to another device, such as a switch or a router, for the physical layer to be active. There are several commands that can be used to verify interface configuration including the **show ip interface brief** and **show ipv6 interface brief**, the **show ip route** and **show ipv6 route**, as well as **show interfaces**, **show ip interface** and **show ipv6 interface.**

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLANs

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Definitions

VLANs are logical connections with other similar devices.

Placing devices into various VLANs have the following characteristics:

- Provides segmentation of the various groups of devices on the same switches
- Provide organization that is more manageable
  - Broadcasts, multicasts and unicasts are isolated in the individual VLAN
  - Each VLAN will have its own unique range of IP addressing
  - Smaller broadcast domains

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Benefits of a VLAN Design

Benefits of using VLANs are as follows:

Benefits	Description
Smaller Broadcast Domains	Dividing the LAN reduces the number of broadcast domains
Improved Security	Only users in the same VLAN can communicate together
Improved IT Efficiency	VLANs can group devices with similar requirements, e.g. faculty vs. students
Reduced Cost	One switch can support multiple groups or VLANs
Better Performance	Small broadcast domains reduce traffic, improving bandwidth
Simpler Management	Similar groups will need similar applications and other network resources

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Types of VLANs

Default VLAN

VLAN 1 is the following:

- The default VLAN
- The default Native VLAN
- The default Management VLAN
- Cannot be deleted or renamed

```
Switch# show vlan brief
VLAN Name                 Status    Ports
---- ----------------- -------  -------------------
1    default              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gi0/1, Gi0/2
1002 fddi-default                        act/unsup
1003 token-ring-default                  act/unsup
1004 fddinet-default                     act/unsup
1005 trnet-default                       act/unsup
```

Note: While we cannot delete VLAN1 Cisco will recommend that we assign these default features to other VLANs

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Types of VLANs (Cont.)

The Native VLAN is a concept used in trunk links (which are used to carry traffic between switches or network devices) and is an important part of the 802.1Q VLAN tagging protocol. Here's a breakdown of its key points:
Trunk Links: A trunk link is used to carry traffic from multiple VLANs between network devices like switches and routers. These links typically use the 802.1Q VLAN tagging standard to distinguish between frames belonging to different VLANs. Each frame traveling over a trunk link is tagged with the VLAN ID that it belongs to.

Data VLAN

- Dedicated to user-generated traffic (email and web traffic).
- VLAN 1 is the default data VLAN because all interfaces are assigned to this VLAN.

Native VLAN

Native VLAN Definition: The Native VLAN is the default VLAN assigned to a trunk link for untagged frames. In other words, if a frame enters the trunk link and does not have any VLAN tag (i.e., it is not tagged as belonging to a specific VLAN), it is automatically assumed to belong to the Native VLAN.

- This is used for trunk links only.
- All frames are tagged on an 802.1Q trunk link except for those on the native VLAN.

Management VLAN

- This is used for SSH/Telnet VTY traffic and should not be carried with end user traffic.
- Typically, the VLAN that is the SVI for the Layer 2 switch.

A Management VLAN is a dedicated VLAN used for network management purposes, such as accessing network devices for configuration, monitoring, and troubleshooting. Here’s a breakdown of its key points:
Purpose: The primary role of the Management VLAN is to handle SSH (Secure Shell) or Telnet traffic for remote management of network devices, such as switches, routers, and firewalls. It ensures that management traffic is kept separate from the regular end-user data traffic.
Traffic Isolation: Since the Management VLAN carries only management-related traffic (SSH, Telnet, SNMP, etc.), it should not be mixed with regular user data traffic. This isolation improves security and prevents accidental interference or congestion with general network activities.

2024 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Defining VLAN Trunks

SVI (Switched Virtual Interface): On Layer 2 switches, the Management VLAN is usually associated with the SVI, which is a virtual interface used for Layer 3 communication. The SVI provides the switch with an IP address, enabling it to be managed remotely through protocols like SSH or Telnet. This VLAN is typically configured with the switch's management IP address.
Security Consideration: To enhance security, network administrators typically place the Management VLAN on a physically separate network or use VLAN Access Control Lists (VACLs) to control which devices or subnets can access the management network.

A trunk is a point-to-point link between two network devices.

Cisco trunk functions:

- Allow more than one VLAN
- Extend the VLAN across the entire network
- By default, supports all VLANs
- Supports 802.1Q trunking

Unlike an access port, which is assigned to a single VLAN, a trunk port can carry traffic for multiple VLANs simultaneously. This is crucial for enabling communication between VLANs across multiple switches.

Trunking allows VLANs to exist across multiple switches by carrying VLAN traffic over a single physical link.
This helps in maintaining VLAN segmentation across the entire network.
Example: If VLAN 10 exists on Switch A and Switch B, a trunk link between them ensures that devices in VLAN 10 can communicate, even though they are connected to different switches.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Networks without VLANs

Without VLANs, all devices connected to the switches will receive all unicast, multicast, and broadcast traffic.

PC1 sends out a local Layer 2 broadcast. The switches forward the broadcast frame out all available ports.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Networks with VLANs

Why is a Layer 3 Device Required for Inter-VLAN Communication?
VLANs (Virtual Local Area Networks) segment network traffic at Layer 2 (Data Link Layer) of the OSI model. This segmentation improves security and efficiency, but it also prevents communication between devices in different VLANs by default.

With VLANs, unicast, multicast, and broadcast traffic is confined to a VLAN. Without a Layer 3 device to connect the VLANs, devices in different VLANs cannot communicate.

Key Point: Layer 2 Switches Cannot Route Between VLANs
A Layer 2 switch forwards Ethernet frames based on MAC addresses and does not have the capability to route packets between different VLANs. Each VLAN acts as a separate broadcast domain, meaning devices within the same VLAN can communicate, but devices in different VLANs cannot communicate unless a Layer 3 device is used

PC1 sends out a local Layer 2 broadcast. The switches forward the broadcast frame only out ports configured for VLAN10.

A Layer 3 device (such as a router or a Layer 3 switch) is required to route traffic between VLANs. This is called Inter-VLAN Routing and can be achieved in two main ways:1. Router-on-a-Stick (Using a Router)
A router with a single trunk link to the switch is used for routing between VLANs. The router has subinterfaces, each assigned an IP address corresponding to the VLANs. The switch sends VLAN-tagged traffic to the router, which then routes it to the correct VLAN.

© 2016 - Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Identification with a Tag

- The IEEE 802.1Q header is 4 Bytes
- When the tag is created the FCS must be recalculated.
- When sent to end devices, this tag must be removed and the FCS recalculated back to its original number.

802.1Q VLAN Tag Field	Function
Type	• 2-Byte field with hexadecimal 0x8100 • This is referred to as Tag Protocol ID (TPID)
User Priority	• 3-bit value that supports
Canonical Format Identifier (CFI)	• 1-bit value that can support token ring frames on Ethernet
VLAN ID (VID)	• 12-bit VLAN identifier that can support up to 4096 VLANs

Frame Check Sequence (FCS) Must Be Recalculated When a Tag is Added
The Frame Check Sequence (FCS) is a CRC (Cyclic Redundancy Check) value used for error detection.
When an 802.1Q VLAN tag is added, the frame length changes, which invalidates the original FCS.
The switch recalculates the FCS and updates it before forwarding the frame.

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Native VLANs and 802.1Q Tagging

The Native VLAN is a special VLAN used on trunk links to handle untagged frames. Whether the Native VLAN has a tag or not depends on how the network is configured:
Default Behavior: Native VLAN Frames Are Untagged
By default, on a trunk link, all VLAN traffic is tagged using 802.1Q VLAN tagging, except for frames belonging to the Native VLAN. Frames in the Native VLAN are sent without any VLAN tag across the trunk.
When the switch receives an untagged frame on a trunk port, it assigns it to the Native VLAN.

802.1Q trunk basics:

- Tagging is typically done on all VLANs.
- The use of a native VLAN was designed for legacy use, like the hub in the example.
- Unless changed, VLAN1 is the native VLAN.
- Both ends of a trunk link must be configured with the same native VLAN.
- Each trunk is configured separately, so it is possible to have a different native VLANs on separate trunks.

Each trunk can have a different Native VLAN because it is a port-level setting.
Be cautious of Native VLAN mismatches between switches.
Consider tagging the Native VLAN to prevent security vulnerabilities.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Configuration

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Configuration

VLAN Ranges on Catalyst Switches

```
Switch# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gi0/1, Gi0/2
1002 fddi-default                                     act/unsup
1003 token-ring-default                               act/unsup
1004 fddinet-default                                  act/unsup
1005 trnet-default                                    act/unsup
```

Catalyst switches 2960 and 3650 support over 4000 VLANs.

Normal Range VLAN 1 – 1005	Extended Range VLAN 1006 - 4095
Used in Small to Medium sized businesses	Used by Service Providers
1002 – 1005 are reserved for legacy VLANs	Are in Running-Config
1, 1002 – 1005 are auto created and cannot be deleted	Supports fewer VLAN features
Stored in the vlan.dat file in flash	Requires VTP configurations
VTP can synchronize between switches	

VTP (VLAN Trunking Protocol) is a Cisco proprietary protocol that synchronizes VLAN configurations across multiple switches in a network. It ensures that VLAN information is consistently updated across all connected switches without manual configuration on each switch.

VLAN Creation Commands

VLAN details are stored in the vlan.dat file. You create VLANs in the global configuration mode.

Task	IOS Command
Enter global configuration mode.	Switch# **configure terminal**
Create a VLAN with a valid ID number.	Switch(config)# **vlan** *vlan-id*
Specify a unique name to identify the VLAN.	Switch(config-vlan)# **name** *vlan-name*
Return to the privileged EXEC mode.	Switch(config-vlan)# **end**
Enter global configuration mode.	Switch# **configure terminal**

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Creation Example

- If the Student PC is going to be in VLAN 20, we will create the VLAN first and then name it.
- If you do not name it, the Cisco IOS will give it a default name of vlan and the four digit number of the VLAN. E.g. vlan0020 for VLAN 20.

Prompt	Command
S1#	Configure terminal
S1(config)#	vlan 20
S1(config-vlan)#	name student
S1(config-vlan)#	end

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Port Assignment Commands

Once the VLAN is created, we can then assign it to the correct interfaces.

Task	Command
Enter global configuration mode.	Switch# **configure terminal**
Enter interface configuration mode.	Switch(config)# **interface** *interface-id*
Set the port to access mode.	Switch(config-if)# **switchport mode access**
Assign the port to a VLAN.	Switch(config-if)# **switchport access vlan** *vlan-id*
Return to the privileged EXEC mode.	Switch(config-if)# **end**

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Port Assignment Example

We can assign the VLAN to the port interface.

- Once the device is assigned the VLAN, then the end device will need the IP address information for that VLAN
- Here, Student PC receives 172.17.20.22

Prompt	Command
S1#	Configure terminal
S1(config)#	Interface fa0/18
S1(config-if)#	Switchport mode access
S1(config-if)#	Switchport access vlan 20
S1(config-if)#	end

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Verify VLAN Information

Use the **show vlan** command. The complete syntax is:

show vlan [**brief** | **id** *vlan-id* | **name** *vlan-name* | **summary**]

```
S1# show vlan summary
Number of existing VLANs              : 7
 Number of existing VTP VLANs         : 7
 Number of existing extended VLANS    : 0
```

```
S1# show interface vlan 20
Vlan20 is up, line protocol is up
  Hardware is EtherSVI, address is 001f.6ddb.3ec1 (bia 001f.6ddb.3ec1)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set

(Output omitted)
```

Task	Command Option
Display VLAN name, status, and its ports one VLAN per line.	**brief**
Display information about the identified VLAN ID number.	**id** *vlan-id*
Display information about the identified VLAN name. The *vlan-name* is an ASCII string from 1 to 32 characters.	**name** *vlan-name*
Display VLAN summary information.	**summary**

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Change VLAN Port Membership

There are a number of ways to change VLAN membership:

- re-enter **switchport access vlan** *vlan-id* command
- use the **no switchport access vlan** to place interface back in VLAN 1

Use the **show vlan brief** or the **show interface fa0/18 switchport** commands to verify the correct VLAN association.

```
S1(config)# interface fa0/18
S1(config-if)# no switchport access vlan
S1(config-if)# end
S1#
S1# show vlan brief
VLAN Name                             Status    Ports
---- -------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                               Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                               Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                               Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                               Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                               Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                               Gi0/1, Gi0/2
20   student                          active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Delete VLANs

Delete VLANs with the **no vlan** *vlan-id* command.

Caution: Before deleting a VLAN, reassign all member ports to a different VLAN.

- Delete all VLANs with the **delete flash:vlan.dat** or **delete vlan.dat** commands.
- Reload the switch when deleting all VLANs.

Note: To restore to factory default – unplug all data cables, erase the startup-configuration and delete the vlan.dat file, then reload the device.

To reset a Cisco switch to factory default settings,

Feature	Access Mode	Trunk Mode
Number of VLANs	One VLAN per port	Multiple VLANs per port
VLAN Tagging	No (untagged)	Yes (802.1Q or ISL)
Used for	End devices (PCs, printers)	Switch-to-switch, switch-to-router
Default VLAN	Assigned manually	Usually VLAN 1 unless changed

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

VLAN Trunks

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Trunk Configuration Commands

A trunk is a Layer 2 link that carries traffic for multiple VLANs over a single physical connection.

How Do Trunks Work?

A trunk extends VLANs across the network by allowing multiple VLANs to pass through a single link.

Configure and verify VLAN trunks. Trunks are layer 2 and carry traffic for all VLANs.

It is commonly used between switches or between a switch and a router. Trunk links use VLAN tagging (802.1Q) to identify VLANs.

Task	IOS Command
Enter global configuration mode.	Switch# **configure terminal**
Enter interface configuration mode.	Switch(config)# **interface** *interface-id*
Set the port to permanent trunking mode.	Switch(config-if)# **switchport mode trunk**
Sets the native VLAN to something other than VLAN 1.	Switch(config-if)# **switchport trunk native vlan** *vlan-id*
Specify the list of VLANs to be allowed on the trunk link.	Switch(config-if)# **switchport trunk allowed vlan** *vlan-list*
Return to the privileged EXEC mode.	Switch(config-if)# **end**

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Trunk Configuration Example

The subnets associated with each VLAN are:

- VLAN 10 - Faculty/Staff - 172.17.10.0/24
- VLAN 20 - Students - 172.17.20.0/24
- VLAN 30 - Guests - 172.17.30.0/24
- VLAN 99 - Native - 172.17.99.0/24

F0/1 port on S1 is configured as a trunk port.

Note: This assumes a 2960 switch using 802.1q tagging. Layer 3 switches require the encapsulation to be configured before the trunk mode.

Prompt	Command
S1(config)#	Interface fa0/1
S1(config-if)#	Switchport mode trunk
S1(config-if)#	Switchport trunk native vlan 99
S1(config-if)#	Switchport trunk allowed vlan 10,20,30,99
S1(config-if)#	end

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Verify Trunk Configuration

Set the trunk mode and native vlan.

Notice **sh int fa0/1 switchport** command:

- Is set to trunk administratively
- Is set as trunk operationally (functioning)
- Encapsulation is dot1q
- Native VLAN set to VLAN 99
- All VLANs created on the switch will pass traffic on this trunk

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode trunk
S1(config-if)# no switchport trunk native vlan 99
S1(config-if)# end
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (VLAN0099)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
(output omitted)
```

802.1Q (Dot1Q) is an industry-standard VLAN tagging protocol. It adds a VLAN tag to Ethernet frames to indicate which VLAN the frame belongs to.
This command is used only on router sub-interfaces or older switch models that support ISL and 802.1Q.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Inter-VLAN Routing

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What is Inter-VLAN Routing?

VLANs are used to segment switched Layer 2 networks for a variety of reasons. Regardless of the reason, hosts in one VLAN cannot communicate with hosts in another VLAN unless there is a router or a Layer 3 switch to provide routing services.

Inter-VLAN routing is the process of forwarding network traffic from one VLAN to another VLAN.

There are three inter-VLAN routing options:

- **Legacy Inter-VLAN routing** - This is a legacy solution. It does not scale well.
- **Router-on-a-Stick** - This is an acceptable solution for a small to medium-sized network.
- **Layer 3 switch using switched virtual interfaces (SVIs)** - This is the most scalable solution for medium to large organizations.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Legacy Inter-VLAN Routing

- The first inter-VLAN routing solution relied on using a router with multiple Ethernet interfaces.
- Each router interface was connected to a switch port in different VLANs.
- However, it is **not scalable** because routers have a limited number of physical interfaces. Requiring **one physical router interface per VLAN**
- **Note**: This method of inter-VLAN routing is no longer implemented and is included for explanation purposes only.

The number of physical interfaces in a router varies by type:

Home/Small Office Routers --> 5 to 6
(1 WAN + 4 LAN + 1 Console)
Branch Routers (e.g., Cisco ISR) --> 4 to 10
Enterprise Routers --> 8 to 32+
Data Center/High-End Routers --> Up to 100+ ports
(modular chassis-based routers)

In general, a router has two or more physical ports, depending on the model. A basic router typically has:
2 to 4 Ethernet ports (e.g., GigabitEthernet0/0, GigabitEthernet0/1)
1 Console port (for management)
1 AUX port (optional, for remote access)
1 or more WAN ports (e.g., Serial0/0/0 on older routers, or GE/WAN ports on modern ones)

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What is Inter-VLAN routing?

- Layer 2 switches cannot forward traffic between VLANs without the assistance of a router.
- Inter-VLAN routing is a process for forwarding network traffic from one VLAN to another, using a router.

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Router-on-a-Stick Inter-VLAN Routing

The ‘router-on-a-stick’ inter-VLAN routing method overcomes the limitation of the legacy inter-VLAN routing method.

It only requires **one physical Ethernet interface to route traffic between multiple VLANs** on a network.

- A Cisco IOS router Ethernet interface is configured as an **802.1Q trunk** and connected to a trunk port on a Layer 2 switch. Specifically,
- The router interface is configured using **subinterfaces** to identify routable VLANs.
- The configured subinterfaces are software-based virtual interfaces.
- Each subinterface is independently configured with an IP address and VLAN assignment.
- When VLAN-tagged traffic enters the router interface, it is forwarded to the VLAN subinterface. After a routing decision is made based on the destination IP network address, the router determines the exit interface for the traffic. If the exit interface is configured as an 802.1q subinterface, the data frames are VLAN-tagged with the new VLAN and sent back out the physical interface

VLAN-tagged traffic ----> router interface ---> VLAN subinterface --->routing decision ---> sending to the destination VLAN subinterface

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Legacy Inter-VLAN Routing

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure Legacy Inter-VLAN Routing

Switch Configuration

The no shutdown command is used in router and switch configurations to enable an interface that is administratively down. It is essential to bring an interface up after configuring it.

```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/11
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/4
S1(config-if)# switchport access vlan 10
S1(config-if)# interface f0/6
S1(config-if)# switchport access vlan 30
S1(config-if)# interface f0/5
S1(config-if)# switchport access vlan 30
S1(config-if)# end
*Mar 20 01:22:56.751: %SYS-5-CONFIG_I: Configured from console by
console
S1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

1. On Router Interfaces
When configuring physical or logical interfaces, you must use no shutdown to activate them.
Example: Enabling a Router Interface

2. On Switch Interfaces (Layer 2 and Layer 3)
On Layer 2 switches: Not usually needed for access ports, but required for trunk ports.
On Layer 3 switches: Required when enabling a routed port.

3. On VLAN Interfaces (SVI - Switch Virtual Interface)
For SVIs, the no shutdown command is needed to activate the virtual interface.

When NOT to Use no shutdown
On end devices (PCs, printers, etc.)—not applicable.
On Layer 2 switch access ports (they are up by default when a device is connected).

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Router Interface Configuration

```
R1(config)# interface g0/0
R1(config-if)# ip address 172.17.10.1 255.255.255.0
R1(config-if)# no shutdown
*Mar 20 01:42:12.951: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to up
*Mar 20 01:42:13.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up
R1(config-if)# interface g0/1
R1(config-if)# ip address 172.17.30.1 255.255.255.0
R1(config-if)# no shutdown
*Mar 20 01:42:54.951: %LINK-3-UPDOWN: Interface GigabitEthernet0/1,
changed state to up
*Mar 20 01:42:55.951: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to up
R1(config-if)# end
R1# copy running-config startup-config
```

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Router-on-a-Stick Inter-VLAN Routing

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Switch Configuration

```
S1(config)# vlan 10
S1(config-vlan)# vlan 30
S1(config-vlan)# interface f0/5
S1(config-if)# switchport mode trunk
S1(config-if)# end
S1#
```

co and/or its affiliates. All rights reserved. Cisco Confidential

Router Subinterface Configuration

```
R1(config)# interface g0/0.10
R1(config-subif)# encapsulation dot1q 10
R1(config-subif)# ip address 172.17.10.1 255.255.255.0
R1(config-subif)# interface g0/0.30
R1(config-subif)# encapsulation dot1q 30
R1(config-subif)# ip address 172.17.30.1 255.255.255.0
R1(config)# interface g0/0
R1(config-if)# no shutdown
*Mar 20 00:20:59.299: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to down
*Mar 20 00:21:02.919: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to up
*Mar 20 00:21:03.919: %LINEPROTO-5-UPDOWN: Line protocol on
 changed state to down
*Mar 20 00:21:02.919: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to up
*Mar 20 00:21:03.919: %LINEPROTO-5-UPDOWN: Line protocol on
 Interface GigabitEthernet0/0, changed state to up
```

ll rights reserved. Cisco Confidential

Inter-VLAN Routing using Layer 3 Switches

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Inter-VLAN Routing on a Layer 3 Switch

The modern method of performing inter-VLAN routing is to use Layer 3 switches and switched virtual interfaces (SVI). An SVI is a virtual interface that is configured on a Layer 3 switch, as shown in the figure.

Note: A Layer 3 switch is also called a multilayer switch as it operates at Layer 2 and Layer 3. However, in this course we use the term Layer 3 switch.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Inter-VLAN Routing on a Layer 3 Switch (Cont.)

Inter-VLAN SVIs are created the same way that the management VLAN interface is configured. The SVI is created for a VLAN that exists on the switch. Although virtual, the SVI performs the same functions for the VLAN as a router interface would. Specifically, it provides Layer 3 processing for packets that are sent to or from all switch ports associated with that VLAN.

The following are advantages of using Layer 3 switches for inter-VLAN routing:

- They are much faster than router-on-a-stick because everything is hardware switched and routed.
- There is no need for external links from the switch to the router for routing.
- They are not limited to one link because Layer 2 EtherChannels can be used as trunk links between the switches to increase bandwidth.
- Latency is much lower because data does not need to leave the switch in order to be routed to a different network.
- They more commonly deployed in a campus LAN than routers.

- The only disadvantage is that Layer 3 switches are more expensive.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Layer 3 Switch Scenario

In the figure, the Layer 3 switch, D1, is connected to two hosts on different VLANs. PC1 is in VLAN 10 and PC2 is in VLAN 20, as shown. The Layer 3 switch will provide inter-VLAN routing services to the two hosts.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Layer 3 Switch Configuration

Complete the following steps to configure S1 with VLANs and trunking:

- **Step 1**. Create the VLANs. In the example, VLANs 10 and 20 are used.
- **Step 2**. Create the SVI VLAN interfaces. The IP address configured will serve as the default gateway for hosts in the respective VLAN. Int vlan 10 ; ip address
- **Step 3**. Configure access ports. Assign the appropriate port to the required VLAN.
- **Step 4**. Enable IP routing. Issue the **ip routing** global configuration command to allow traffic to be exchanged between VLANs 10 and 20. This command must be configured to enable inter-VAN routing on a Layer 3 switch for IPv4.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Routing on a Layer 3 Switch

If VLANs are to be reachable by other Layer 3 devices, then they must be advertised using static or dynamic routing. To enable routing on a Layer 3 switch, a routed port must be configured.

A routed port is created on a Layer 3 switch by disabling the switchport feature on a Layer 2 port that is connected to another Layer 3 device. Specifically, configuring the **no switchport** interface configuration command on a Layer 2 port converts it into a Layer 3 interface. Then the interface can be configured with an IPv4 configuration to connect to a router or another Layer 3 switch.

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Network Address Translation

Dr. Md. Shohrab Hossain
Professor, CSE, BUET

Source: Cisco Networking

IPv4 Address Space

- Networks are commonly implemented using private IPv4 addresses, as defined in RFC 1918.
- Private IPv4 addresses cannot be routed over the internet and are used within an organization or site to allow devices to communicate locally.
- To allow a device with a private IPv4 address to access devices and resources outside of the local network, the private address must first be translated to a public address.
- NAT provides the translation of private addresses to public addresses.

Class	Activity Type	Activity Name
A	10.0.0.0 – 10.255.255.255	10.0.0.0/8
B	172.16.0.0 – 172.31.255.255	172.16.0.0/12
C	192.168.0.0 – 192.168.255.255	192.168.0.0/16

What is NAT

- The primary use of NAT is to conserve public IPv4 addresses.
- NAT allows networks to use private IPv4 addresses internally and translates them to a public address when needed.
- A NAT router typically operates at the border of a stub network.
- When a device inside the stub network wants to communicate with a device outside of its network, the packet is forwarded to the border router which performs the NAT process, translating the internal private address of the device to a public, outside, routable address.

How NAT Works

PC1 wants to communicate with an outside web server with public address 209.165.201.1.

1. PC1 sends a packet addressed to the web server.
2. R2 receives the packet and reads the source IPv4 address to determine if it needs translation.
3. R2 adds mapping of the local to global address to the NAT table.
4. R2 sends the packet with the translated source address toward the destination.
5. The web server responds with a packet addressed to the inside global address of PC1 (209.165.200.226).
6. R2 receives the packet with destination address 209.165.200.226. R2 checks the NAT table and finds an entry for this mapping. R2 uses this information and translates the inside global address (209.165.200.226) to the inside local address (192.168.10.10), and the packet is forwarded toward PC1.

NAT Table

Inside Local	Inside Global	Outside Local	Outside Global
192.168.10.10	209.165.200.226	209.165.201.1	209.165.201.1

NAT Terminology

NAT includes four types of addresses:

- Inside local address
- Inside global address
- Outside local address
- Outside global address

NAT terminology is always applied from the perspective of the device with the translated address:

- **Inside address** - The address of the device which is being translated by NAT.
- **Outside address** - The address of the destination device.
- **Local address** - A local address is any address that appears on the inside portion of the network.
- **Global address** - A global address is any address that appears on the outside portion of the network.

NAT Terminology (Cont.)

Inside local address

The address of the source as seen from inside the network. This is typically a private IPv4 address. The inside local address of PC1 is 192.168.10.10.

Inside global addresses

The address of source as seen from the outside network. The inside global address of PC1 is 209.165.200.226

Outside global address

The address of the destination as seen from the outside network. The outside global address of the web server is 209.165.201.1

Outside local address

The address of the destination as seen from the inside network. PC1 sends traffic to the web server at the IPv4 address 209.165.201.1. While uncommon, this address could be different than the globally routable address of the destination.

Policy-Based NAT (Customized NAT Mapping)
A company may translate an external server's real IP into an alias IP specific to their network for security or routing reasons.
Example: A public website 198.51.100.20 might be accessed internally as 192.168.200.1

R2 NAT Table

PC1		Web Server	
Inside Global Address	Inside Local Address	Outside Local Address	Outside Global Address
209.165.200.226	192.168.10.10	209.165.201.1	209.165.201.1

When Can "Outside Local Address" Be Different?
Private Network-to-Private Network Translation (Double NAT)
If both internal and external networks use private IPs, the outside local address may be different from the actual outside global address.
Example: A company using 10.1.1.1 to reach an external server, but the real IP of the server is 172.16.1.1.

VPN or MPLS Networks
If a VPN or MPLS system remaps public IPs to internal ones,
the outside local address can be different from the real public IP of the destination.

Types of NAT

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Static NAT

Static NAT uses a one-to-one mapping of local and global addresses configured by the network administrator that remain constant.

- Static NAT is useful for web servers or devices that must have a consistent address that is accessible from the internet, such as a company web server.
- It is also useful for devices that must be accessible by authorized personnel when offsite, but not by the general public on the internet.

Static NAT Table

Inside Local Address	Inside Global Address - Addresses reachable via R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

Note: Static NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

Dynamic NAT

Dynamic NAT uses a pool of public addresses and assigns them on a first-come, first-served basis.

- When an inside device requests access to an outside network, dynamic NAT assigns an available public IPv4 address from the pool.
- The other addresses in the pool are still available for use.

Note: Dynamic NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

IPv4 NAT Pool

Inside Local Address	Inside Global Address Pool - Addresses reachable via R2
192.168.10.12	209.165.200.226
Available	209.165.200.227
Available	209.165.200.228
Available	209.165.200.229
Available	209.165.200.230

Port Address Translation

Port Address Translation (PAT), also known as NAT overload, maps multiple private IPv4 addresses to a single public IPv4 address or a few addresses.

- With PAT, when the NAT router receives a packet from the client, it uses the source port number to uniquely identify the specific NAT translation.
- PAT ensures that devices use a different TCP port number for each session with a server on the internet.

NAT Table with Overload

Inside Local IP Address	Inside Global IP Address	Outside Local IP Address	Outside Global IP Address
192.168.10.10:1555	209.165.200.226:1555	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1331	209.165.200.226:1331	209.165.202.129:80	209.165.202.129:80

Next Available Port

PAT attempts to preserve the original source port. If the original source port is already used, PAT assigns the first available port number starting from the beginning of the appropriate port group 0-511, 512-1,023, or 1,024-65,535.

- When there are no more ports available and there is more than one external address in the address pool, PAT moves to the next address to try to allocate the original source port.
- The process continues until there are no more available ports or external IPv4 addresses in the address pool.

NAT and PAT Comparison

Summary of the differences between NAT and PAT.

NAT - Only modifies the IPv4 addresses

Inside Global Address	Inside Local Address
209.165.200.226	192.168.10.10

PAT - PAT modifies both the IPv4 address and the port number.

Inside Global Address	Inside Local Address
209.165.200.226:2031	192.168.10.10:2031

NAT	PAT
One-to-one mapping between Inside Local and Inside Global addresses.	One Inside Global address can be mapped to many Inside Local addresses.
Uses only IPv4 addresses in translation process.	Uses IPv4 addresses and TCP or UDP source port numbers in translation process.
A unique Inside Global address is required for each inside host accessing the outside network.	A single unique Inside Global address can be shared by many inside hosts accessing the outside network.

Advantages of NAT

(conserves legally registered addressing scheme, conserves addresses, increases flexibility of connections, provides consistency, hides the IPv4 addresses of users and other devices)

NAT provides many benefits:

- NAT conserves the legally registered addressing scheme by allowing the privatization of intranets.
- NAT conserves addresses through application port-level multiplexing.
- NAT increases the flexibility of connections to the public network.
- NAT provides consistency for internal network addressing schemes.
- NAT allows the existing private IPv4 address scheme to remain while allowing for easy change to a new public addressing scheme.
- NAT hides the IPv4 addresses of users and other devices.

Disadvantages of NAT

(delay, end-to-end addressing is lost, traceability is lost, complicates the use of tunneling protocols, Services using UDP can be disrupted)

NAT does have drawbacks:

- NAT increases forwarding delays.
- End-to-end addressing is lost.
- End-to-end IPv4 traceability is lost.
- NAT complicates the use of tunneling protocols, such as IPsec.
- Services that require the initiation of TCP connections from the outside network, or stateless protocols, such as those using UDP, can be disrupted.

Static NAT

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Static NAT Scenario

- Static NAT is a one-to-one mapping between an inside address and an outside address.
- Static NAT allows external devices to initiate connections to internal devices using the statically assigned public address.
- For instance, an internal web server may be mapped to a specific inside global address so that it is accessible from outside networks.

Configure Static NAT

There are two basic tasks when configuring static NAT translations:

- **Step 1** - Create a mapping between the inside local address and the inside global addresses using the **ip nat inside source static** command.
- **Step 2** - The interfaces participating in the translation are configured as inside or outside relative to NAT with the **ip nat inside** and **ip nat outside** commands.

```
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5
R2(config)#
R2(config)# interface serial 0/1/0
R2(config-if)# ip address 192.168.1.2 255.255.255.252
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface serial 0/1/1
R2(config-if)# ip address 209.165.200.1 255.255.255.252
R2(config-if)# ip nat outside
```

Analyze Static NAT

The static NAT translation process between the client and the web server:

1. The client sends a packet to the web server.
2. R2 receives packets from the client on its NAT outside interface and checks its NAT table.
3. R2 translates the inside global address of to the inside local address and forwards the packet towards the web server.
4. The web server receives the packet and responds to the client using its inside local address.
5. (a) R2 receives the packet from the web server on its NAT inside interface with source address of the inside local address of the web server and (b) translates the source address to the inside global address.

	Inside Local Address	Inside Global Address	Outside Local Address	Outside Global Address
2 5b	192.168.10.254	209.165.201.5	209.165.200.254	209.165.200.254

Verify Static NAT

To verify NAT operation, issue the **show ip nat translations** command.

- This command shows active NAT translations.
- Because the example is a static NAT configuration, the translation is always present in the NAT table regardless of any active communications.
- If the command is issued during an active session, the output also indicates the address of the outside device.

```
R2# show ip nat translations
Pro  Inside global       Inside local       Outside local     Outside global
---  209.165.201.5       192.168.10.254     ---               ---
Total number of translations: 1
```

```
R2# show ip nat translations
Pro  Inside global       Inside local        Outside local         Outside global
tcp  209.165.201.5       192.168.10.254      209.165.200.254       209.165.200.254
---  209.165.201.5       192.168.10.254        ---                   ---
Total number of translations: 2
```

Verify Static NAT (Cont.)

Another useful command is **show ip nat statistics.**

- It displays information about the total number of active translations, NAT configuration parameters, the number of addresses in the pool, and the number of addresses that have been allocated.
- To verify that the NAT translation is working, it is best to clear statistics from any past translations using the **clear ip nat statistics** command before testing.

```
R2# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Outside interfaces:
  Serial0/1/1
Inside interfaces:
  Serial0/1/0
Hits: 4  Misses: 1
(output omitted)
```

Dynamic NAT

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Dynamic NAT Scenario

- Dynamic NAT automatically maps inside local addresses to inside global addresses.
- Dynamic NAT uses a pool of inside global addresses.
- The pool of inside global addresses is available to any device on the inside network on a first-come first-served basis.
- If all addresses in the pool are in use, a device must wait for an available address before it can access the outside network.

Analyze Dynamic NAT – Inside to Outside

ALC (Access List Configuration)

Dynamic NAT translation process:

1. PC1 and PC2 send packets requesting a connection to the server.
2. R2 receives the first packet from PC1, checks the ALC to determine if the packet should be translated, selects an available global address, and creates a translation entry in the NAT table.
3. R2 replaces the inside local source address of PC1, 192.168.10.10, with the translated inside global address of 209.165.200.226 and forwards the packet. (The same process occurs for the packet from PC2 using the translated address of 209.165.200.227.)

IPv4 NAT Pool	
Inside Local Address Pool	Inside Global Address
192.168.10.10	209.165.200.226
192.168.11.10	209.165.200.227

Configure Dynamic NAT

There are five tasks when configuring dynamic NAT translations:

- **Step 1** - Define the pool of addresses that will be used for translation using the **ip nat pool** command.
- **Step 2** - Configure a standard ACL to identify (permit) only those addresses that are to be translated.
- **Step 3** - Bind the ACL to the pool, using the **ip nat inside source list** command.

(Access Control Lists)

```
R2(config)# ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL1
```

Configure Dynamic NAT (Cont.)

There are five tasks when configuring dynamic NAT translations:

- **Step 4** - Identify which interfaces are inside.
- **Step 5** - Identify which interfaces are outside.

```
R2(config)# ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL1
R2(config)# interface serial 0/1/0
R2(config-if)# ip nat inside
R2(config-if)# interface serial 0/1/1
R2(config-if)# ip nat outside
```

Analyze Dynamic NAT – Outside to Inside

Dynamic NAT translation process:

4. The server receives the packet from PC1 and responds using the destination address of 209.165.200.226. The server receives the packet from PC2, it responds to using the destination address of 209.165.200.227.
5. (a) When R2 receives the packet with the destination address of 209.165.200.226; it performs a NAT table lookup and translates the address back to the inside local address and forwards the packet toward PC1.
   (b) When R2 receives the packet with the destination address of 209.165.200.227; it performs a NAT table lookup and translates the address back to the inside local address 192.168.11.10 and forwards the packet toward PC2.

	IPv4 NAT Pool	
	Inside Local Address Pool	**Inside Global Address**
5a	192.168.10.10	209.165.200.226
5b	192.168.11.10	209.165.200.227

Verify Dynamic NAT

The output of the **show ip nat translations** command displays all static translations that have been configured and any dynamic translations that have been created by traffic.

```
R2# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 209.165.200.228    192.168.10.10      ---                ---
--- 209.165.200.229    192.168.11.10      ---                ---
R2#
```

Verify Dynamic NAT (Cont.)

By default, translation entries time out after 24 hours, unless the timers have been reconfigured with the **ip nat translation timeout** *timeout-seconds* command in global configuration mode. To clear dynamic entries before the timeout has expired, use the **clear ip nat translation** privileged EXEC mode command.

```
R2# clear ip nat translation *
R2# show ip nat translation
```

Command	Description
clear ip nat translation *	Clears all dynamic address translation entries from the NAT translation table.
clear ip nat translation inside *global-ip local-ip* [**outside** *local-ip global-ip*]	Clears a simple dynamic translation entry containing an inside translation or both inside and outside translation.
clear ip nat translation *protocol* **inside** *global-ip global-port local-ip local-port* [**outside** *local-ip local-port global-ip global-port*]	Clears an extended dynamic translation entry.

PAT

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Configure PAT to Use a Single IPv4 Address

To configure PAT to use a single IPv4 address, add the keyword **overload** to the **ip nat inside source** command.

In the example, all hosts from network 192.168.0.0/16 (matching ACL 1) that send traffic through router R2 to the internet will be translated to IPv4 address 209.165.200.225 (IPv4 address of interface S0/1/1). The traffic flows will be identified by port numbers in the NAT table because the **overload** keyword is configured.

```
R2(config)# ip nat inside source list 1 interface serial 0/1/0 overload
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# interface serial0/1/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface Serial0/1/1
R2(config-if)# ip nat outside
```

Configure PAT to Use an Address Pool

An ISP may allocate more than one public IPv4 address to an organization. In this scenario the organization can configure PAT to use a pool of IPv4 public addresses for translation.

To configure PAT for a dynamic NAT address pool, simply add the keyword **overload** to the **ip nat inside source** command.

In the example, NAT-POOL2 is bound to an ACL to permit 192.168.0.0/16 to be translated. These hosts can share an IPv4 address from the pool because PAT is enabled with the keyword **overload**.

```
R2(config)# ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask 255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL2 overload
R2(config)# interface serial0/1/0
R2(config-if)# ip nat inside
R2(config-if)# interface serial0/1/0
R2(config-if)# ip nat outside
```

Analyze PAT – Server to PC

1. PC1 and PC2 send packets to Svr1 and Svr2.
2. The packet from PC1 reaches R2 first. R2 modifies the source IPv4 address to 209.165.200.225 (inside global address). The packet is then forwarded towards Svr1.
3. The packet from PC2 arrives at R2. PAT changes the source IPv4 address of PC2 to the inside global address 209.165.200.225. PC2 has the same source port number as the translation for PC1. PAT increments the source port number until it is a unique value in its table. In this instance, 1445.

NAT Table

Inside Local Address	Inside Global Address	Outside Global Address	Outside Local Address
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.201.129:80	209.165.202.129:80

Analyze PAT – PC to Server

1. PC1 and PC2 send packets to Svr1 and Svr2.
2. The packet from PC1 reaches R2 first. R2 modifies the source IPv4 address to 209.165.200.225 (inside global address). The packet is then forwarded towards Svr1.
3. The packet from PC2 arrives at R2. PAT changes the source IPv4 address of PC2 to the inside global address 209.165.200.225. PC2 has the same source port number as the translation for PC1. PAT increments the source port number until it is a unique value in its table. In this instance, it is 1445.

NAT Table

Inside Local Address	Inside Global Address	Outside Global Address	Outside Local Address
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.201.129:80	209.165.202.129:80

Analyze PAT – Server to PC

1. The servers use the source port from the received packet as the destination port, and the source address as the destination address for the return traffic.
2. R2 changes the destination IPv4 address of the packet from Srv1 from 209.165.200.225 to 192.168.10.10, and forwards the packet toward PC1.
3. R2 changes the destination address of packet from Srv2. from 209.165.200.225 to 192.168.10.11. and modifies the destinations port back to its original value of 1444. The packet is then forwarded toward PC2.

NAT Table

Inside Local Address	Inside Global Address	Outside Global Address	Outside Local Address
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.201.129:80	209.165.202.129:80

Verify PAT

The same commands used to verify static and dynamic NAT are used to verify PAT. The **show ip nat translations** command displays the translations from two different hosts to different web servers. Notice that two different inside hosts are allocated the same IPv4 address of 209.165.200.226 (inside global address). The source port numbers in the NAT table differentiate the two transactions.

```
R2# show ip nat translations
Pro Inside global          Inside local         Outside local      Outside global
tcp 209.165.200.225:1444  192.168.10.10:1444  209.165.201.1:80   209.165.201.1:80
tcp 209.165.200.225:1445  192.168.11.10:1444  209.165.202.129:80 209.165.202.129:80
R2#
```

Access Control Lists

Source: Cisco Networking

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What is an ACL?

An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header. By default, a router does not have any ACLs configured. When an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded.

An ACL uses a sequential list of permit or deny statements, known as access control entries (ACEs).

Note: ACEs are also commonly called ACL statements.

When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering.

What is an ACL? (Cont.)

Several tasks performed by routers require the use of ACLs to identify traffic:

- Limit network traffic to increase network performance
- Provide traffic flow control
- Provide a basic level of security for network access
- Filter traffic based on traffic type
- Screen hosts to permit or deny access to network services
- Provide priority to certain classes of network traffic

Packet Filtering

Packet filtering controls access to a network by analyzing the incoming and/or outgoing packets and forwarding them or discarding them based on given criteria.

Packet filtering can occur at Layer 3 or Layer 4.

Cisco routers support two types of ACLs:

- **Standard ACLs** - ACLs only filter at Layer 3 using the source IPv4 address only.
- **Extended ACLs** - ACLs filter at Layer 3 using the source and / or destination IPv4 address. They can also filter at Layer 4 using TCP, UDP ports, and optional protocol type information for finer control.

ACL Operation

ACLs define the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router, and packets that exit outbound interfaces of the router.

ACLs can be configured to apply to inbound traffic and outbound traffic.

Note: ACLs do not act on packets that originate from the router itself.

An inbound ACL filters packets before they are routed to the outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet is discarded.

An outbound ACL filters packets after being routed, regardless of the inbound interface.

ACL Operation (Cont.)

When an ACL is applied to an interface, it follows a specific operating procedure. Here are the operational steps used when traffic has entered a router interface with an inbound standard IPv4 ACL configured:

1. The router extracts the source IPv4 address from the packet header.
2. The router starts at the top of the ACL and compares the source IPv4 address to each ACE in a sequential order.
3. When a match is made, the router carries out the instruction, either permitting or denying the packet, and the remaining ACEs in the ACL, if any, are not analyzed.
4. If the source IPv4 address does not match any ACEs in the ACL, the packet is discarded because there is an implicit deny ACE automatically applied to all ACLs.

The last ACE statement of an ACL is always an implicit deny that blocks all traffic. It is hidden and not displayed in the configuration.

Note: An ACL must have at least one permit statement otherwise all traffic will be denied due to the implicit deny ACE statement.

Wildcard Mask Overview

A wildcard mask is similar to a subnet mask in that it uses the ANDing process to identify which bits in an IPv4 address to match. Unlike a subnet mask, in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask, the reverse is true.

An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match.

Wildcard masks use the following rules to match binary 1s and 0s:

- **Wildcard mask bit 0** - Match the corresponding bit value in the address
- **Wildcard mask bit 1** - Ignore the corresponding bit value in the address

Purpose of the Wildcard Mask
It is used in ACLs (Access Control Lists), OSPF, and EIGRP to match specific IP ranges.
The inversion helps simplify calculations when checking IP address matches.

The wildcard mask is reversed from the subnet mask because it serves a different function: matching specific IP address patterns in ACLs and routing protocols, rather than defining network boundaries

Wildcard Mask Overview (Cont.)

Wildcard Mask	Last Octet (in Binary)	Meaning (0 - match, 1 - ignore)
0.0.0.0	00000000	Match all octets.
0.0.0.63	00111111	• Match the first three octets • Match the two left most bits of the last octet • Ignore the last 6 bits
0.0.0.15	00001111	• Match the first three octets • Match the four left most bits of the last octet • Ignore the last 4 bits of the last octet
0.0.0.248	11111100	• Match the first three octets • Ignore the six left most bits of the last octet • Match the last two bits
0.0.0.255	11111111	• Match the first three octet • Ignore the last octet

Wildcard Mask Types

access-list 10
The number 10 is the Access Control List (ACL) number.
In standard ACLs, numbers range from 1 to 99 (or 1300 to 1999 in extended range).
This means it is a Standard ACL, which only filters traffic based on the source IP address.

Wildcard to Match a Host:

Assume ACL 10 needs an ACE that only permits the host with IPv4 address 192.168.1.1. Recall that “0” equals a match and “1” equals ignore. To match a specific host IPv4 address, a wildcard mask consisting of all zeroes (i.e., 0.0.0.0) is required.

When the ACE is processed, the wildcard mask will permit only the 192.168.1.1 address. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.1 0.0.0.0.**

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.0	00000000.00000000.00000000.00000000
Permitted IPv4 Address	**192.168.1.1**	11000000.10101000.00000001.00000001

Wildcard Mask Types (Cont.)

Wildcard Mask to Match an IPv4 Subnet

ACL 10 needs an ACE that permits all hosts in the 192.168.1.0/24 network. The wildcard mask 0.0.0.255 stipulates that the very first three octets must match exactly but the fourth octet does not.

When processed, the wildcard mask 0.0.0.255 permits all hosts in the 192.168.1.0/24 network. The resulting ACE in ACL 10 would be

access-list 10 permit 192.168.1.0 0.0.0.255.

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111
Permitted IPv4 Address	**192.168.1.0/24**	11000000.10101000.00000001.00000000

Wildcard Mask Types (Cont.)

Wildcard Mask to Match an IPv4 Address Range

ACL 10 needs an ACE that permits all hosts in the 192.168.16.0/24, 192.168.17.0/24, …, 192.168.31.0/24 networks.

When processed, the wildcard mask 0.0.15.255 permits all hosts in the 192.168.16.0/24 to 192.168.31.0/24 networks. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.16.0 0.0.15.255.**

	Decimal	Binary
IPv4 address	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111
Permitted IPv4 Address	**192.168.16.0/24 to 192.168.31.0/24**	11000000.10101000.00010000.00000000 11000000.10101000.00011111.00000000

Guidelines for ACL Creation

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Limited Number of ACLs per Interface

There is a limit on the number of ACLs that can be applied on a router interface. For example, a dual-stacked (i.e, IPv4 and IPv6) router interface can have up to four ACLs applied, as shown in the figure.

Specifically, a router interface can have:

- One outbound IPv4 ACL.
- One inbound IPv4 ACL.
- One inbound IPv6 ACL.
- One outbound IPv6 ACL.

Note: ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the security policy of the organization.

ACL Best Practices

Using ACLs requires attention to detail and great care. Mistakes can be costly in terms of downtime, troubleshooting efforts, and poor network service. Basic planning is required before configuring an ACL.

Guideline	Benefit
Base ACLs on the organizational security policies.	This will ensure you implement organizational security guidelines.
Write out what you want the ACL to do.	This will help you avoid inadvertently creating potential access problems.
Use a text editor to create, edit, and save all of your ACLs.	This will help you create a library of reusable ACLs.
Document the ACLs using the **remark** command.	This will help you (and others) understand the purpose of an ACE.
Test the ACLs on a development network before implementing them on a production network.	This will help you avoid costly errors.

Types of IPv4 ACLs

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Standard and Extended ACLs

There are two types of IPv4 ACLs:

- **Standard ACLs** - These permit or deny packets based only on the source IPv4 address.
- **Extended ACLs** - These permit or deny packets based on the source IPv4 address and destination IPv4 address, protocol type, source and destination TCP or UDP ports and more.

Numbered and Named ACLs

Numbered ACLs

ACLs numbered 1-99, or 1300-1999 are standard ACLs, while ACLs numbered 100-199, or 2000-2699 are extended ACLs.

```
R1(config)# access-list ?
 <1-99> IP standard access list
 <100-199> IP extended access list
 <1100-1199> Extended 48-bit MAC address access list
 <1300-1999> IP standard access list (expanded range)
 <200-299> Protocol type-code access list
 <2000-2699> IP extended access list (expanded range)
 <700-799> 48-bit MAC address access list
 rate-limit Simple rate-limit specific access list
 template Enable IP template acls
Router(config)# access-list
```

Numbered and Named ACLs (Cont.)

Named ACLs

Named ACLs are the preferred method to use when configuring ACLs. Specifically, standard and extended ACLs can be named to provide information about the purpose of the ACL. For example, naming an extended ACL FTP-FILTER is far better than having a numbered ACL 100.

The **ip access-list** global configuration command is used to create a named ACL, as shown in the following example.

```
R1(config)# ip access-list extended FTP-FILTER
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data
R1(config-ext-nacl)#
```

Where to Place ACLs

Every ACL should be placed where it has the greatest impact on efficiency.

Extended ACLs should be located as close as possible to the source of the traffic to be filtered.

Standard ACLs should be located as close to the destination as possible.

Extended ACLs filter traffic based on source IP, destination IP, and protocols (ports, services, etc.).
Reason:
If the unwanted traffic is filtered near the source, it is blocked early, preventing unnecessary traffic from traveling through the network. This reduces bandwidth usage and improves network efficiency.

Standard ACLs only filter based on source IP, so they cannot specify the destination.
Reason:
If placed too close to the source, it could block traffic to all destinations, even unintended ones. By placing it near the destination, only traffic reaching that specific network or device is filtered.

Where to Place ACLs (Cont.)

If a Standard ACL is placed too close to the source, it blocks all traffic from that source, no matter the destination.
Instead, placing it near the destination ensures that only traffic to that specific destination is blocked, while allowing access to everything else.

Factors Influencing ACL Placement	Explanation
The extent of organizational control	Placement of the ACL can depend on whether or not the organization has control of both the source and destination networks.
Bandwidth of the networks involved	It may be desirable to filter unwanted traffic at the source to prevent transmission of bandwidth-consuming traffic.
Ease of configuration	• It may be easier to implement an ACL at the destination, but traffic will use bandwidth unnecessarily. • An extended ACL could be used on each router where the traffic originated. This would save bandwidth by filtering the traffic at the source, but it would require creating extended ACLs on multiple routers.

Standard ACL Placement Example

In the figure, the administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.

Following the basic placement guidelines, the administrator would place a standard ACL on router R3.

Standard ACL Placement Example (Cont.)

There are two possible interfaces on R3 to apply the standard ACL:

- **R3 S0/1/1 interface (inbound)** - The standard ACL can be applied inbound on the R3 S0/1/1 interface to deny traffic from .10 network. However, it would also filter .10 traffic to the 192.168.31.0/24 (.31 in this example) network. Therefore, the standard ACL should not be applied to this interface.
- **R3 G0/0 interface (outbound)** - The standard ACL can be applied outbound on the R3 G0/0/0 interface. This will not affect other networks that are reachable by R3. Packets from .10 network will still be able to reach the .31 network. This is the best interface to place the standard ACL to meet the traffic requirements.

Extended ACL Placement Example

Extended ACLs should be located as close to the source as possible.

However, the organization can only place ACLs on devices that they control. Therefore, the extended ACL placement must be determined in the context of where organizational control extends.

In the figure, for example, Company A wants to deny Telnet and FTP traffic to Company B's 192.168.30.0/24 network from their 192.168.11.0/24 network, while permitting all other traffic.

Extended ACL Placement Example (Cont.)

An extended ACL on R3 would accomplish the task, but the administrator does not control R3. In addition, this solution allows unwanted traffic to cross the entire network, only to be blocked at the destination.

The solution is to place an extended ACL on R1 that specifies both source and destination addresses.

There are two possible interfaces on R1 to apply the extended ACL:

- **R1 S0/1/0 interface (outbound)** - The extended ACL can be applied outbound on the S0/1/0 interface. This solution will process all packets leaving R1 including packets from 192.168.10.0/24.
- **R1 G0/0/1 interface (inbound)** - The extended ACL can be applied inbound on the G0/0/1 and only packets from the 192.168.11.0/24 network are subject to ACL processing on R1. Because the filter is to be limited to only those packets leaving the 192.168.11.0/24 network, applying the extended ACL to G0/1 is the best solution.

Configure Standard IPv4 ACLs

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Create an ACL

All access control lists (ACLs) must be planned. When configuring a complex ACL, it is suggested that you:

- Use a text editor and write out the specifics of the policy to be implemented.
- Add the IOS configuration commands to accomplish those tasks.
- Include remarks to document the ACL.
- Copy and paste the commands onto the device.
- Always thoroughly test an ACL to ensure that it correctly applies the desired policy.

Numbered Standard IPv4 ACL Syntax

To create a numbered standard ACL, use the **access-list** command.

```
Router(config)# access-list access-list-number {deny | permit | remark text} source [source-wildcard]
[log]
```

Parameter	Description
access-list-number	Number range is 1 to 99 or 1300 to 1999
deny	Denies access if the condition is matched
permit	Permits access if the condition is matched
remark *text*	(Optional) text entry for documentation purposes
source	Identifies the source network or host address to filter
source-wildcard	(Optional) 32-bit wildcard mask that is applied to the source
log	(Optional) Generates and sends an informational message when the ACE is matched

Note: Use the **no access-list** *access-list-number* global configuration command to remove a numbered standard ACL.

Numbered Standard ACL Example

The example ACL permits traffic from host 192.168.10.10 and all hosts on the 192.168.20.0/24 network out interface serial 0/1/0 on router R1.

```
R1(config)# access-list 10 remark ACE permits ONLY host 192.168.10.10 to the internet
R1(config)# access-list 10 permit host 192.168.10.10
R1(config)# do show access-lists
Standard IP access list 10
    10 permit 192.168.10.10
R1(config)#
```

```
R1(config)# access-list 10 remark ACE permits all host in LAN 2
R1(config)# access-list 10 permit 192.168.20.0 0.0.0.255
R1(config)# do show access-lists
Standard IP access list 10
    10 permit 192.168.10.10
    20 permit 192.168.20.0, wildcard bits 0.0.0.255
R1(config)#
```

```
R1(config)# interface Serial 0/1/0
R1(config-if)# ip access-group 10 out
R1(config-if)# end
R1#
```

Numbered Standard ACL Example (Cont.)

Use the **show running-config** command to review the ACL in the configuration.

Use the **show ip interface** command to verify the ACL is applied to the interface.

```
R1# show run | section access-list
access-list 10 remark ACE permits host 192.168.10.10
access-list 10 permit 192.168.10.10
access-list 10 remark ACE permits all host in LAN 2
access-list 10 permit 192.168.20.0 0.0.0.255
R1#
```

```
R1# show ip int Serial 0/1/0 | include access list
  Outgoing Common access list is not set
  Outgoing access list is 10
  Inbound Common access list is not set
  Inbound  access list is not set
R1#
```

Named Standard ACL Example

The example ACL permits traffic from host 192.168.10.10 and all hosts on the 192.168.20.0/24 network out interface serial 0/1/0 on router R1.

```
R1(config)# no access-list 10
R1(config)# ip access-list standard PERMIT-ACCESS
R1(config-std-nacl)# remark ACE permits host 192.168.10.10
R1(config-std-nacl)# permit host 192.168.10.10
R1(config-std-nacl)#
```

```
R1(config-std-nacl)# remark ACE permits host 192.168.10.10
R1(config-std-nacl)# permit host 192.168.10.10
R1(config-std-nacl)# remark ACE permits all hosts in LAN 2
R1(config-std-nacl)# permit 192.168.20.0 0.0.0.255
R1(config-std-nacl)# exit
R1(config)#
```

```
R1(config)# interface Serial 0/1/0
R1(config-if)# ip access-group PERMIT-ACCESS out
R1(config-if)# end
R1#
```

Two Methods to Modify an ACL

After an ACL is configured, it may need to be modified. ACLs with multiple ACEs can be complex to configure. Sometimes the configured ACE does not yield the expected behaviors.

There are two methods to use when modifying an ACL:

- Use a text editor.
- Use sequence numbers.

Text Editor Method

ACLs with multiple ACEs should be created in a text editor. This allows you to plan the required ACEs, create the ACL, and then paste it into the router interface. It also simplifies the tasks to edit and fix an ACL.

To correct an error in an ACL:

- Copy the ACL from the running configuration and paste it into the text editor.
- Make the necessary edits or changes.
- Remove the previously configured ACL on the router.
- Copy and paste the edited ACL back to the router.

```
R1# show run | section access-list
access-list 1 deny   19.168.10.10
access-list 1 permit 192.168.10.0 0.0.0.255
R1#
```

```
R1(config)# no access-list 1
R1(config)#
R1(config)# access-list 1 deny 192.168.10.10
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#
```

Sequence Number Method

An ACL ACE can be deleted or added using the ACL sequence numbers.

- Use the **ip access-list standard** command to edit an ACL.
- Statements cannot be overwritten using an existing sequence number. The current statement must be deleted first with the **no 10** command. Then the correct ACE can be added using sequence number.

```
R1# show access-lists
Standard IP access list 1
    10 deny   19.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Secure VTY with Standard ACL

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Secure VTY Access Example

This example demonstrates how to configure an ACL to filter vty traffic.

- First, a local database entry for a user **ADMIN** and password **class** is configured.
- The vty lines on R1 are configured to use the local database for authentication, permit SSH traffic, and use the ADMIN-HOST ACL to restrict traffic.

```
R1(config)# username ADMIN secret class
R1(config)# ip access-list standard ADMIN-HOST
R1(config-std-nacl)# remark This ACL secures incoming vty lines
R1(config-std-nacl)# permit 192.168.10.10
R1(config-std-nacl)# deny any
R1(config-std-nacl)# exit
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input telnet
R1(config-line)# access-class ADMIN-HOST in
R1(config-line)# end
R1#
```

Verify the VTY Port is Secured

After an ACL to restrict access to the vty lines is configured, it is important to verify it works as expected.

To verify the ACL statistics, issue the **show access-lists** command.

- The match in the permit line of the output is a result of a successful SSH connection by host with IP address 192.168.10.10.
- The match in the deny statement is due to the failed attempt to create a SSH connection from a device on another network.

```
R1#
Oct  9 15:11:19.544: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 192.168.10.10]
[localport: 23] at 15:11:19 UTC Wed Oct 9 2019
R1# show access-lists
Standard IP access list ADMIN-HOST
    10 permit 192.168.10.10  (2 matches)
    20 deny   any  (2 matches)
R1#
```

Configure Extended IPv4 ACLs

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Extended ACLs

Extended ACLs provide a greater degree of control. They can filter on source address, destination address, protocol (i.e., IP, TCP, UDP, ICMP), and port number.

Extended ACLs can be created as:

- **Numbered Extended ACL** - Created using the **access-list** *access-list-number* global configuration command.
- **Named Extended ACL** - Created using the **ip access-list extended** *access-list-name*.

Protocols and Ports (Cont.)

Selecting a protocol influences port options. Many TCP port options are available, as shown in the output.

```
R1(config)# access-list 100 permit tcp any any eq ?
  <0-65535>    Port number
  bgp          Border Gateway Protocol (179)
  chargen      Character generator (19)
  cmd          Remote commands (rcmd, 514)
  daytime      Daytime (13)
  discard      Discard (9)
  domain       Domain Name Service (53)
  echo         Echo (7)
  exec         Exec (rsh, 512)
  finger       Finger (79)
  ftp          File Transfer Protocol (21)
  ftp-data     FTP data connections (20)
  gopher       Gopher (70)
  hostname     NIC hostname server (101)
  ident        Ident Protocol (113)
  irc          Internet Relay Chat (194)
  klogin       Kerberos login (543)
  kshell       Kerberos shell (544)
  login        Login (rlogin, 513)
  lpd          Printer service (515)
  msrpc        MS Remote Procedure Call (135)
  nntp         Network News Transport Protocol (119)
  onep-plain   Onep Cleartext (15001)
  onep-tls     Onep TLS (15002)
  pim-auto-rp  PIM Auto-RP (496)
  pop2         Post Office Protocol v2 (109)
  pop3         Post Office Protocol v3 (110)
  smtp         Simple Mail Transport Protocol (25)
  sunrpc       Sun Remote Procedure Call (111)
  syslog       Syslog (514)
  tacacs       TAC Access Control System (49)
  talk         Talk (517)
  telnet       Telnet (23)
  time         Time (37)
  uucp         Unix-to-Unix Copy Program (540)
  whois        Nicname (43)
  www          World Wide Web (HTTP, 80)
```

Apply a Numbered Extended IPv4 ACL

In this example, the ACL permits both HTTP and HTTPS traffic from the 192.168.10.0 network to go to any destination.

Extended ACLs can be applied in various locations. However, they are commonly applied close to the source. Here ACL 110 is applied inbound on the R1 G0/0/0 interface.

```
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 110 in
R1(config-if)# exit
R1(config)#
```

TCP Established Extended ACL

TCP can also perform basic stateful firewall services using the TCP **established** keyword.

- The **established** keyword enables inside traffic to exit the inside private network and permits the returning reply traffic to enter the inside private network.
- TCP traffic generated by an outside host and attempting to communicate with an inside host is denied.

TCP Established Extended ACL (Cont.)

ACL 120 is configured to only permit returning web traffic to the inside hosts. The ACL is then applied outbound on the R1 G0/0/0 interface.

The **show access-lists** command shows that inside hosts are accessing the secure web resources from the internet.

Note: A match occurs if the returning TCP segment has the ACK or reset (RST) flag bits set, indicating that the packet belongs to an existing connection.

```
R1(config)# access-list 120 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 120 out
R1(config-if)# end
R1# show access-lists
Extended IP access list 110
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (657 matches)
Extended IP access list 120
    10 permit tcp any 192.168.10.0 0.0.0.255 established (1166 matches)
R1#
```

Named Extended IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. To create a named extended ACL, use the **ip access-list extended** configuration command.

In the example, a named extended ACL called NO-FTP-ACCESS is created and the prompt changed to named extended ACL configuration mode. ACE statements are entered in the named extended ACL sub configuration mode.

```
Router(config)# ip access-list extended access-list-name
```

```
R1(config)# ip access-list extended NO-FTP-ACCESS
R1(config-ext-nacl)#
```

Named Extended IPv4 ACL Example

The topology below is used to demonstrate configuring and applying two named extended IPv4 ACLs to an interface:

SURFING - This will permit inside HTTP and HTTPS traffic to exit to the internet.

BROWSING - This will only permit returning web traffic to the inside hosts while all other traffic exiting the R1 G0/0/0 interface is implicitly denied.

Named Extended IPv4 ACL Example (Cont.)

The SURFING ACL permits HTTP and HTTPS traffic from inside users to exit the G0/0/1 interface connected to the internet. Web traffic returning from the internet is permitted back into the inside private network by the BROWSING ACL.

The SURFING ACL is applied inbound and the BROWSING ACL is applied outbound on the R1 G0/0/0 interface.

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# Remark Permits inside HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# Remark Only permit returning HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
R1(config-if)# end
R1# show access-lists
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (124 matches)
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established (369 matches)
R1#
```

Named Extended IPv4 ACL Example (Cont.)

The show access-lists command is used to verify the ACL statistics. Notice that the permit secure HTTPS counters (i.e., eq 443) in the SURFING ACL and the return established counters in the BROWSING ACL have increased.

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 19.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Another Extended IPv4 ACL Example

Two named extended ACLs will be created:

PERMIT-PC1 - This will only permit PC1 TCP access to the internet and deny all other hosts in the private network.

REPLY-PC1 - This will only permit specified returning TCP traffic to PC1 implicitly deny all other traffic.

Another Extended IPv4 ACL Example (Cont.)

The **PERMIT-PC1** ACL permits PC1 (192.168.10.10) TCP access to the FTP, SSH, Telnet, DNS , HTTP, and HTTPS traffic.

The **REPLY-PC1** ACL will permit return traffic to PC1.

The **PERMIT-PC1** ACL is applied inbound and the **REPLY-PC1** ACL applied outbound on the R1 G0/0/0 interface.

```
R1(config)# ip access-list extended PERMIT-PC1
R1(config-ext-nacl)# Remark Permit PC1 TCP access to internet
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 20
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 21
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 22
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 23
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 53
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 80
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 443
R1(config-ext-nacl)# deny ip 192.168.10.0 0.0.0.255 any
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended REPLY-PC1
R1(config-ext-nacl)# Remark Only permit returning traffic to PC1
R1(config-ext-nacl)# permit tcp any host 192.168.10.10 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group PERMIT-PC1 in
R1(config-if)# ip access-group REPLY-PC1 out
R1(config-if)# end
R1#
```

Verify Extended ACLs

The **show ip interface** command is used to verify the ACL on the interface and the direction in which it was applied.

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up (connected)
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is REPLY-PC1
  Inbound  access list is PERMIT-PC1
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is disabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
R1#
R1# show ip interface g0/0/0 | include access list
Outgoing access list is REPLY-PC1
Inbound access list is PERMIT-PC1
R1#
```

Verify Extended ACLs (Cont.)

The **show access-lists** command can be used to confirm that the ACLs work as expected. The command displays statistic counters that increase whenever an ACE is matched.

Note: Traffic must be generated to verify the operation of the ACL.

```
R1# show access-lists
Extended IP access list PERMIT-PC1
10 permit tcp host 192.168.10.10 any eq 20
20 permit tcp host 192.168.10.10 any eq ftp
30 permit tcp host 192.168.10.10 any eq 22
40 permit tcp host 192.168.10.10 any eq telnet
50 permit tcp host 192.168.10.10 any eq domain
60 permit tcp host 192.168.10.10 any eq www
70 permit tcp host 192.168.10.10 any eq 443
80 deny ip 192.168.10.0 0.0.0.255 any
Extended IP access list REPLY-PC1
10 permit tcp any host 192.168.10.10 established
R1#
```

Network Monitoring and Wireshark

CSE 322

Acknowledgement

- https://studylib.net/doc/17803129/cap6135--malware-and-software-vulnerability-analysis-netw...
- https://slideplayer.com/slide/5692294/
- http://ilta.ebiz.uapps.net/ProductFiles/productfiles/672/**wireshark.ppt**

- UC Berkley course "EE 122: Intro to Communication Networks"
  - http://www.eecs.berkeley.edu/~jortiz/courses/ee122/presentations/Wireshark.ppt
- Other resources:
  - http://openmaniak.com/wireshark_filters.php

Motivation for Network Monitoring

- Essential for Network Management
  - Router and Firewall policy
  - Detecting abnormal/error in networking
  - Access control
- Security Management
  - Detecting abnormal traffic
  - Traffic log for future forensic analysis

Tools Overview

- Tcpdump
  - Unix-based command-line tool used to intercept packets
    - Including filtering just the packets of interest
  - Reads "live traffic" from interface specified using **`-i`** option …
  - … or from a previously recorded *trace file* specified using **`-r`** option
    - You create these when capturing live traffic using **`-w`** option
- Tshark
  - Tcpdump-like capture program
  - Very similar behavior and flags to tcpdump
- Wireshark
  - GUI for displaying tcpdump/tshark packet traces

tcpdump Example

- Ran tcpdump on a Unix machine
- Example: first few lines of the output

```
01:46:28.808262 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-
    7.dsl.pltn13.pacbell.net.2481: . 2513546054:2513547434(1380) ack
    1268355216 win 12816
01:46:28.808271 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-
    7.dsl.pltn13.pacbell.net.2481: P 1380:2128(748) ack 1 win 12816
01:46:28.808276 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-
    7.dsl.pltn13.pacbell.net.2481: . 2128:3508(1380) ack 1 win 12816
01:46:28.890021 IP adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481 >
    danjo.CS.Berkeley.EDU.ssh: P 1:49(48) ack 1380 win 16560
```

What Does a Line Convey?

Timestamp

This is a TCP packet

Source host name

Source port number (22)

01:46:28.808262 IP danjo.CS.Berkeley.EDU.ssh >
adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481: .
2513546054:2513547434(1380) ack 1268355216 win 12816

Destination host name

Destination port number

TCP specific information

- Different output formats for different packet types

Output from tcpdump

```
manav@ubuntulinux: ~
manav@ubuntulinux:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
23:12:15.734637 IP6 fe80::6c8:7ff:fe26:ceac > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
23:12:15.737024 IP ubuntulinux.60811 > b.resolvers.Level3.net.domain: 60873+ PTR? 6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
23:12:15.937343 IP b.resolvers.Level3.net.domain > ubuntulinux.60811: 60873 NXDomain 0/1/0 (154)
23:12:15.939727 IP ubuntulinux.44132 > b.resolvers.Level3.net.domain: 7027+ PTR? c.a.e.c.6.2.e.f.f.f.7.0.8.c.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
23:12:16.142628 IP b.resolvers.Level3.net.domain > ubuntulinux.44132: 7027 NXDomain* 0/1/0 (149)
23:12:16.144477 IP ubuntulinux.54078 > b.resolvers.Level3.net.domain: 44074+ PTR? 2.2.2.4.in-addr.arpa. (38)
23:12:16.182985 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QU)? Android.local. ANY (QU)? Android.local. (81)
23:12:16.346940 IP b.resolvers.Level3.net.domain > ubuntulinux.54078: 44074 1/0/0 PTR b.resolvers.Level3.net. (74)
23:12:16.348070 IP ubuntulinux.40932 > b.resolvers.Level3.net.domain: 34612+ PTR? 102.0.168.192.in-addr.arpa. (44)
23:12:16.442544 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? Android.local. ANY (QM)? Android.local. (81)
23:12:16.515363 IP b.resolvers.Level3.net.domain > ubuntulinux.40932: 34612 NXDomain* 0/1/0 (103)
23:12:16.517489 IP ubuntulinux.53519 > b.resolvers.Level3.net.domain: 3094+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
23:12:16.590778 IP6 fe80::6c8:7ff:fe26:ceac > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
23:12:16.680420 IP b.resolvers.Level3.net.domain > ubuntulinux.53519: 3094 NXDomain 0/1/0 (154)
23:12:16.683273 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? Android.local. ANY (QM)? Android.local. (81)
23:12:16.961783 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0*- [0q] 4/0/3 (Cache flush) PTR Android.local., (Cache flush) PTR Android.local., (Cache flush) A 192.168.0.101, (Cache flush) AAAA fe80::6c8:7f
f:fe26:ceac (247)
23:12:17.986627 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0*- [0q] 4/0/3 (Cache flush) PTR Android.local., (Cache flush) PTR Android.local., (Cache flush) A 192.168.0.101, (Cache flush) AAAA fe80::6c8:7f
f:fe26:ceac (247)
23:12:18.238252 IP ubuntulinux.35076 > a23-39-122-85.deploy.static.akamaitechnologies.com.https: Flags [.], ack 89677080, win 501, options [nop,nop,TS val 2010432084 ecr 2402112534], length 0
23:12:18.239048 IP ubuntulinux.55784 > b.resolvers.Level3.net.domain: 49805+ PTR? 85.122.39.23.in-addr.arpa. (43)
23:12:18.497830 IP a23-39-122-85.deploy.static.akamaitechnologies.com.https > ubuntulinux.35076: Flags [R], seq 89677080, win 0, length 0
23:12:18.497830 IP b.resolvers.Level3.net.domain > ubuntulinux.55784: 49805 1/0/0 PTR a23-39-122-85.deploy.static.akamaitechnologies.com. (107)
23:12:19.101841 IP ubuntulinux.55857 > 239.255.255.250.1900: UDP, length 171
23:12:19.102591 IP ubuntulinux.49879 > b.resolvers.Level3.net.domain: 54682+ PTR? 250.255.255.239.in-addr.arpa. (46)
23:12:19.317004 IP b.resolvers.Level3.net.domain > ubuntulinux.49879: 54682 NXDomain 0/1/0 (103)
23:12:19.636296 IP ubuntulinux.bootpc > _gateway.bootps: BOOTP/DHCP, Request from 84:fd:d1:e5:20:5e (oui Unknown), length 289
23:12:19.637071 IP ubuntulinux.51783 > b.resolvers.Level3.net.domain: 29099+ PTR? 1.0.168.192.in-addr.arpa. (42)
23:12:19.831442 IP _gateway.bootps > 255.255.255.255.bootpc: BOOTP/DHCP, Reply, length 295
23:12:19.931402 IP b.resolvers.Level3.net.domain > ubuntulinux.51783: 29099 NXDomain* 0/1/0 (101)
23:12:20.102446 IP ubuntulinux.55857 > 239.255.255.250.1900: UDP, length 171
^C23:12:20.239751 IP 192.168.0.3.mdns > 224.0.0.251.mdns: 16 [2q] PTR (QM)? _233637DE._sub._googlecast._tcp.local. PTR (QM)? _googlecast._tcp.local. (61)

30 packets captured
34 packets received by filter
0 packets dropped by kernel
manav@ubuntulinux:~$
```

- Link: https://www.geeksforgeeks.org/tcpdump-command-in-linux-with-examples/

Similar Output from Tshark

1190003744.940437 61.184.241.230 -> 128.32.48.169 SSH Encrypted request packet len=48

1190003744.940916 128.32.48.169 -> 61.184.241.230 SSH Encrypted response packet len=48

1190003744.955764 61.184.241.230 -> 128.32.48.169 TCP 6943 > ssh [ACK] Seq=48 Ack=48 Win=65514 Len=0 TSV=445871583 TSER=632535493

1190003745.035678 61.184.241.230 -> 128.32.48.169 SSH Encrypted request packet len=48

1190003745.036004 128.32.48.169 -> 61.184.241.230 SSH Encrypted response packet len=48

1190003745.050970 61.184.241.230 -> 128.32.48.169 TCP 6943 > ssh [ACK] Seq=96 Ack=96 Win=65514 Len=0 TSV=445871583 TSER=632535502

Demo 1 – Basic Run

- Syntax:

  tcpdump [options] [filter expression]

- To do -
  - $ sudo tcpdump –i eth0
    - Sudo command allows you to run tcpdump in root privilege
  - On your own Unix machine, you can run it using "sudo" or directly run "tcpdump" if you have root privilege
- Observe the output

Filters

- We are often not interested in all packets flowing through the network
- Use filters to capture only packets of interest to us

Demo 2

1. Capture only udp packets
   - tcpdump “udp”
2. Capture only tcp packets
   - tcpdump “tcp”

Demo 2 (contd.)

1. Capture only UDP packets with destination port 53 (DNS requests)
   - tcpdump “udp dst port 53”
2. Capture only UDP packets with source port 53 (DNS replies)
   - tcpdump “udp src port 53”
3. Capture only UDP packets with source or destination port 53 (DNS requests and replies)
   - tcpdump “udp port 53”

Demo 2 (contd.)

1. Capture only packets destined to buet.ac.bd
   - tcpdump “dst host buet.ac.bd”
2. Capture both DNS packets and TCP packets to/from buet.ac.bd
   - tcpdump “(tcp and host buet.ac.bd) or udp port 53”

How to write filters

- Refer the tcpdump/tshark man page
- Many example webpages on the Internet

Running tcpdump

- Requires superuser/administrator privileges on Unix
  - http://www.tcpdump.org/
  - You can do it on your own Unix machine
  - You can install a Linux OS in Vmware on your windows machine

- Tcpdump for Windows
  - WinDump: http://www.winpcap.org/windump/
    - Free software
    - Might have compatibility issues

So What is WireShark?

- Packet sniffer/protocol analyzer
- Open Source Network Tool
- Latest version of the ethereal tool

#	Wireshark	Tcpdump
1	A GUI tool to catch data packets	A CLI-based packet capturing tool
2	It does packet analysis, decode data payloads if the encryption keys are identified, and recognize data payloads from file transfers such as smtp, http, etc.	Tcpdump only provides do a simple analysis of such types of traffic, such as DNS queries
3	It has advanced network interfaces	It has conventional interfaces
4	Good for complex filters	Used for simple filters
5	It provides decoding of protocol-based packet capturing	It is less efficient in decoding compared to Wireshark

tShark
(An Unconventional Alternative)

- Command-line based packet capture tool
- Equivalent to Wireshark

Scope of WireShark

- Internet and all …

Wireshark Interface

Wireshark Interface – Different Parts

Status Bar

Wiredump.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
5031 25.675905 192.168.2.5 103.94.135.200 TCP 66 [TCP Dup ACK 4989#17]
5032 25.675912 103.94.135.200 192.168.2.5 TCP 1514 443 → 49643 [ACK] Seq
5033 25.675926 192.168.2.5 103.94.135.200 TCP 66 [TCP Dup ACK 4989#18]
5034 25.675945 192.168.2.5 103.94.135.200 TLSv1.2 571 Client Hello
5035 25.676037 103.94.135.200 192.168.2.5 TCP 1514 443 → 49643 [ACK] Seq
Frame 5031: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: IntelCor_07:90:b8 (58:91:cf:07:90:b8), Dst: BelkinIn_ea:c6:d5 (94:44:52:ea:c6:d5)
Internet Protocol Version 4, Src: 192.168.2.5, Dst: 103.94.135.200
Transmission Control Protocol, Src Port: 49643, Dst Port: 443, Seq: 1318, Ack: 578795, Len: 0
0000 94 44 52 ea c6 d5 58 91 cf 07 90 b8 08 00 45 00
0010 00 34 44 98 40 00 80 06 04 58 c0 a8 02 05 67 5e
0020 87 c8 c1 eb 01 bb 8b 87 ed ef 4f 36 57 b9 80 10
0030 ff 37 cd dd 00 00 01 01 05 0a 4f 37 8b b1 4f 37
0040 ec a5
Transmission Control Protocol (tcp), 32 bytes
Packets: 30180 · Displayed: 30180 (100.0%)
Profile: Default

Capture Options

- Promiscuous mode is used to capture all traffic
- Sometime this does not work
  - Driver does not support
  - You are on a switched LAN

Capture Filter

Capture Filter Examples

host 10.1.11.24

host 192.168.0.1 and host 10.1.11.1

tcp port http

ip

not broadcast not multicast

ether host 00:04:13:00:09:a3

Capture Buffer Usage

Capture Options

- Ring buffer
  - Addresses a common issue many analysts encounter when capturing packets - huge traces
  - When you use a Ring Buffer, you can define how many files you want to capture and various parameters that affects the file size (# of packets, bytes, and time)
- Link: https://www.networkcomputing.com/networking/working-ring-buffer-wireshark-files

Capture Interfaces

Interface Details

Display Filters (Post-Filters)

- Display filters (also called post-filters) only filter the view of what you are seeing
  - In analysis
  - All packets in the capture still exist in the trace
- Display filters use their own format and are much more powerful then capture filters

Display Filter

Display Filter Examples

ip.src==10.1.11.00/24

ip.addr==192.168.1.10 && ip.addr==192.168.1.20

tcp.port==80 || tcp.port==3389

!(ip.addr==192.168.1.10 && ip.addr==192.168.1.20)

(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (tcp.port==445
    || tcp.port==139)

(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (udp.port==67
    || udp.port==68)

tcp.dstport == 80

Display Filter

Syntax:	Protocol	.	String 1	.	String 2	Comparison operator	Value	Logical Operations	Other expression
Example:	ftp		passive		ip	==	10.2.3.4	xor	icmp.type

TCP Segment Structure

Display Filter

- String1, String2 (Optional settings):
  - Sub protocol categories inside the protocol.
  - Look for a protocol and then click on the "+" character.
  - Example:
  - **tcp.srcport == 80**
  - **tcp.flags == 2**
    - **SYN packet**
    - **Tcp.flags.syn==1**
  - **tcp.flags == 18**
    - **SYN/ACK**

- **TCP Flag field:**

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

Display Filter Expressions

- snmp || dns || icmp
  - Display the SNMP or DNS or ICMP traffics.
- tcp.port == 25
  - Display packets with TCP source or destination port 25.
- tcp.flags
  - Display packets having a TCP flag.
- tcp.flags == 0x02
  - Display packets with a TCP SYN flag.

Six comparison operators are available:

English format:	C like format:	Meaning:
eq	==	Equal
ne	!=	Not equal
gt	>	Greater than
lt	<	Less than
ge	>=	Greater or equal
le	<=	Less or equal

Logical expressions:

English format:	C like format:	Meaning:
and	&&	Logical AND
or	||	Logical OR
xor	^^	Logical XOR
not	!	Logical NOT

If the filter syntax is correct, it will be highlighted in green, otherwise if there is a syntax mistake it will be highlighted in red.

Correct syntax

Wrong syntax

Save Filtered Packets after Using Display Filter

- We can also save all filtered packets in text file for further analysis
- Operation:

File→Export packet dissections →as "plain text" file

1). In "packet range" option, select "Displayed"

2). Choose "summary line" or "detail"

Protocol Hierarchy

Protocol Hierarchy (contd.)

Wireshark · Protocol Hierarchy Statistics · Wiredump.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	30180	100.0	26368284	5445 k	0	0	0
Ethernet	100.0	30180	1.6	422520	87 k	0	0	0
Internet Protocol Version 6	0.0	9	0.0	360	74	0	0	0
User Datagram Protocol	0.0	3	0.0	24	4	0	0	0
DHCPv6	0.0	3	0.0	261	53	3	261	53
Internet Control Message Protocol v6	0.0	6	0.0	192	39	6	192	39
Internet Protocol Version 4	99.9	30160	2.3	603228	124 k	0	0	0
User Datagram Protocol	8.9	2691	0.1	21528	4445	0	0	0
Simple Service Discovery Protocol	0.2	63	0.1	16611	3430	63	16611	3430
QUIC IETF	8.3	2496	5.8	1517524	313 k	2439	1471948	303 k
Malformed Packet	0.0	2	0.0	0	0	2	0	0
NetBIOS Name Service	0.1	23	0.0	1240	256	23	1240	256
NetBIOS Datagram Service	0.0	4	0.0	765	157	0	0	0
SMB (Server Message Block Protocol)	0.0	4	0.0	437	90	0	0	0
SMB MailSlot Protocol	0.0	4	0.0	100	20	0	0	0
Microsoft Windows Browser Protocol	0.0	4	0.0	93	19	4	93	19
Multicast Domain Name System	0.0	4	0.0	244	50	4	244	50
Dynamic Host Configuration Protocol	0.0	3	0.0	900	185	3	900	185
Domain Name System	0.3	95	0.1	14561	3006	95	14561	3006
Data	0.2	58	0.0	2278	470	58	2278	470
Transmission Control Protocol	91.0	27457	90.2	23775639	4909 k	20838	15200882	3139 k
Transport Layer Security	22.1	6658	82.3	21701465	4481 k	6513	18459368	3812 k

No display filter.

Close Copy Help

Follow TCP Stream

Follow TCP Stream

Red - stuff you sent

Blue - stuff you get

Old version

Filter out/in Single TCP Stream

- When click “filter out this TCP stream” in previous page’s box, new filter string will contain like:
  - http and !(tcp.stream eq 5)
- So, if you use “tcp.stream eq 5” as filter string, you keep this HTTP session

Old version

Expert Info

Wiredump.pcap

File Edit View Go Capture Analyze Statistics Telephony

Apply a display filter ... <Ctrl-/>

No.	Time	Source
3	0.000556	23.21
4	0.000606	192.1
5	0.000634	192.1
6	0.014648	192.1

Frame 5: 54 bytes on wire
Ethernet II, Src: IntelCo
Internet Protocol Version
Transmission Control Prot

```
0000  94 44 52 ea c6 d5 58 91  cf 07 90 b8 08 00 45 00
0010  00 28 3c 90 40 00 80 06  e3 ac c0 a8 02 05 17 d5
0020  00 11 c1 88 00 50 70 ac  8a 11 1e f2 00 d9 50 10
0030  10 3b e8 a4 00 00
```

Wiredump.pcap | Packets:

- Display Filters...
- Display Filter Macros...
- Display Filter Expression...
- Apply as Column Ctrl
- Apply as Filter
- Prepare as Filter
- Conversation Filter
- Enabled Protocols... Ctrl
- Decode As... Ctrl
- Reload Lua Plugins Ctrl
- SCTP
- Follow
- Show Packet Bytes... Ctrl
- Expert Information

Wireshark · Expert Information · Wiredump.pcap

Severity	Summary	Group	Protocol
Error	TLSCiphertext length MUST NOT exceed 2^14 + 2048	Protocol	TLS
Error	Malformed Packet (Exception occurred)	Malformed	TLS
Error	Vector length 185 is too large, truncating it to 7	Malformed	TLS
Error	C:\gitlab-builds\builds\-fyeYoMP\1\wireshark\wireshark\...	Malformed	TLS
Warning	DNS query retransmission. Original request in frame 19250	Protocol	DNS
Warning	Connection reset (RST)	Sequence	TCP
Warning	ACKed segment that wasn't captured (common at capture...	Sequence	TCP
Warning	This frame is a (suspected) out-of-order segment	Sequence	TCP
Warning	Ignored Unknown Record	Protocol	TLS
Warning	TCP Zero Window segment	Sequence	TCP
Warning	TCP window specified by the receiver is now completely full	Sequence	TCP
Warning	Previous segment(s) not captured (common at capture sta...	Sequence	TCP
Warning	D-SACK Sequence	Sequence	TCP
Warning	Failed to create decryption context: Secrets are not available	Decryption	QUIC
Note	This frame is a (suspected) fast retransmission	Sequence	TCP
Note	"Time To Live" != 255 for a packet sent to the Local Netwo...	Sequence	IPv4
Note	Duplicate ACK (#1)	Sequence	TCP
Note	This frame is a (suspected) spurious retransmission	Sequence	TCP
Note	This frame is a (suspected) retransmission	Sequence	TCP
Note	This session reuses previously negotiated keys (Session res...	Sequence	TLS
Note	"Time To Live" only 4	Sequence	IPv4

No display filter set.

Limit to Display Filter | Group by summary | Search: | Show...

Close | Help

Conversations

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
0.0.0.0	255.255.255.255	2	684	2	684	0	0	5.878732	1.5431
10.16.15.101	239.255.255.250	9	1503	9	1503	0	0	9.422061	20.7094
10.16.15.101	224.0.0.251	4	412	4	412	0	0	9.886292	22.0151
23.213.0.17	192.168.2.5	6	509	4	401	2	108	0.000000	0.0684
192.168.2.1	224.0.0.1	3	126	3	126	0	0	2.842429	31.3846
192.168.2.1	192.168.2.255	2	102	2	102	0	0	6.680797	30.2073
192.168.2.1	255.255.255.255	1	342	1	342	0	0	7.676869	0.0000
192.168.2.1	239.255.255.250	50	16 k	50	16 k	0	0	8.876733	28.9630
192.168.2.2	224.0.0.251	1	46	1	46	0	0	6.359456	0.0000
192.168.2.5	192.168.2.1	98	19 k	48	3813	50	15 k	0.014648	35.7187
192.168.2.5	192.168.2.255	27	3139	27	3139	0	0	0.450531	23.1789
192.168.2.5	239.255.255.250	5	617	5	617	0	0	1.911801	35.2172
192.168.2.5	142.250.71.10	62	4936	32	2316	30	2620	2.530019	36.1528
192.168.2.5	224.0.0.252	2	92	2	92	0	0	4.130036	30.4989
192.168.2.5	44.192.202.103	18	4987	9	1330	9	3657	4.819424	1.2507
192.168.2.5	224.0.0.251	2	92	2	92	0	0	5.130067	29.9987
192.168.2.5	106.10.218.155	12	3147	7	2593	5	554	5.603879	25.1488
192.168.2.5	119.161.10.12	24	11 k	12	2987	12	8062	5.655297	25.1066
192.168.2.5	142.250.71.46	64	23 k	31	12 k	33	11 k	5.770414	10.5443
192.168.2.5	106.10.236.37	13	1472	7	1019	6	453	5.796128	24.9658

- Use the "Copy" button to copy all text into clipboard (file)
  - Analyze the file to get what statistics you want

Find EndPoint Statistics

- You can sort by field
- Use the "Copy" button to copy all text into clipboard (file)
  - Analyze the file to get what statistics you want

Flow Graphs

- "Limit to display filter" option enable showing the flow of packets displayed in the main interface
- "Export" enables exporting the flow graph shown above

Export HTTP

Wiredump.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Open Ctrl+O
Open Recent
Merge...
Import from Hex Dump...
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Packet Bytes... Ctrl+Shift+X
Export PDUs to File...
Export TLS Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
DICOM...
HTTP...
IMF...
SMB...
TFTP...
Protocol Length Info
Wireshark · Export · HTTP object list
Text Filter:
Content Type: All Content-Types
Packet Hostname Content Type Size Filename
2 text/plain 14 bytes
423 google.com text/html 219 bytes \
1672 www.buet.ac.bd text/html 150 bytes web
Save Save All Preview Close Help

HTTP Analysis

HTTP Analysis – Load Distribution

Wireshark · Load Distribution · Wiredump.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
HTTP Responses by Server Address	6				0.0002	100%	0.0100	0.000
23.213.0.17	1				0.0000	16.67%	0.0100	0.000
OK	1				0.0000	100.00%	0.0100	0.000
192.168.2.1	3				0.0001	50.00%	0.0100	1.917
OK	3				0.0001	100.00%	0.0100	1.917
142.250.182.142	1				0.0000	16.67%	0.0100	9.990
OK	1				0.0000	100.00%	0.0100	9.990
103.94.135.200	1				0.0000	16.67%	0.0100	23.858
OK	1				0.0000	100.00%	0.0100	23.858
HTTP Requests by Server	62				0.0021	100%	0.1600	9.024
HTTP Requests by Server Address	62				0.0021	100.00%	0.1600	9.024
239.255.255.250	60				0.0020	96.77%	0.1600	9.024
239.255.255.250:1900	60				0.0020	100.00%	0.1600	9.024
142.250.182.142	1				0.0000	1.61%	0.0100	9.912
google.com	1				0.0000	100.00%	0.0100	9.912
103.94.135.200	1				0.0000	1.61%	0.0100	23.855
www.buet.ac.bd	1				0.0000	100.00%	0.0100	23.855
HTTP Requests by HTTP Host	62				0.0021	100.00%	0.1600	9.024
www.buet.ac.bd	1				0.0000	1.61%	0.0100	23.855
103.94.135.200	1				0.0000	100.00%	0.0100	23.855
google.com	1				0.0000	1.61%	0.0100	9.912

Display filter: Apply

Copy Save as... Close

HTTP Analysis – Requests

Wireshark · Requests · Wiredump.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
HTTP Requests by HTTP Host	62				0.0021	100%	0.1600	9.024
www.buet.ac.bd	1				0.0000	1.61%	0.0100	23.855
/web/	1				0.0000	100.00%	0.0100	23.855
google.com	1				0.0000	1.61%	0.0100	9.912
/	1				0.0000	100.00%	0.0100	9.912
239.255.255.250:1900	60				0.0020	96.77%	0.1600	9.024
*	60				0.0020	100.00%	0.1600	9.024

Display filter: Apply

Copy Save as... Close

HTTP Analysis – Packet Counter

Wireshark · Packet Counter · Wiredump.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total HTTP Packets	68				0.0023	100%	0.1700	9.990
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	6				0.0002	8.82%	0.0100	0.000
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	2				0.0001	33.33%	0.0100	9.990
301 Moved Permanently	2				0.0001	100.00%	0.0100	9.990
2xx: Success	4				0.0001	66.67%	0.0100	0.000
200 OK	4				0.0001	100.00%	0.0100	0.000
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	62				0.0021	91.18%	0.1600	9.024
SEARCH	12				0.0004	19.35%	0.0100	1.912
NOTIFY	48				0.0016	77.42%	0.1600	9.024
GET	2				0.0001	3.23%	0.0100	9.912

Display filter: Apply

Copy Save as... Close

Improving WireShark Performance

- Don’t use capture filters
- Increase your read buffer size
- Get a faster computer
- Don’t resolve names

Post-Processing Text File

- For saved text-format packet files, further analysis needs coding or special tools
- One useful tool on Unix: Grep
  - On Windows: PowerGrep http://www.powergrep.com/
  - Command-line based utility for searching plain-text data sets for lines matching a regular expression

Basic usage of Grep

- Command-line text-search program in Linux
- Some useful usage:
  - Grep ‘word’ filename # find lines with ‘word’
  - Grep –v ‘word’ filename # find lines without ‘word’
  - Grep ‘^word’ filename # find lines beginning with ‘word’
  - Grep ‘word’ filename > file2 # output lines with ‘word’ to file2
  - ls -l | grep rwxrwxrwx # list files that have ‘rwxrwxrwx’ feature
  - grep '^[0-4]‘ filename # find lines beginning with any of the numbers from 0-4
  - Grep –c ‘word’ filename # find lines with ‘word’ and print out the number of these lines
  - Grep –i ‘word’ filename # find lines with ‘word’ regardless of case
- Many tutorials on grep online
  - http://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/
  - http://www.thegeekstuff.com/2009/03/15-practical-unix-grep-command-examples/

On-line Wireshark Trace Files

- Public available .pcap files:
  - http://www.netresec.com/?page=PcapFiles

- http://www.tp.org/jay/nwanalysis/traces/Lab%20Trace%20Files/

- Wiki Sample capture
  - https://wiki.wireshark.org/SampleCaptures

Example Trace File and Questions

- Network Forensic Puzzle Contests
  - http://forensicscontest.com/2010/02/03/puzzle-4-the-curious-mr-x

- SharkFest'15 Packet Challenge
  - https://sharkfest.wireshark.org/assets/presentations15/packetchallenge.zip

Source of Installer

- https://www.wireshark.org/download.html

Thank You

Acknowledgement: Web sources

Advanced Topics in Networking

CSE 322

At A Glance …

- An advanced execution
- An advanced command
- An advanced tool

An Advanced Execution

- TCP Congestion Control Algorithms (CCAs) in Linux
  - Source: https://jasonmurray.org/posts/2021/tcpcca/

- Check the current CCA
  - sysctl net.ipv4.tcp_congestion_control

Output Example:

```
jemurray@home-server:~$ sysctl net.ipv4.tcp_congestion_control
net.ipv4.tcp_congestion_control = cubic
```

TCP CCA in Linux

- Check for the default CCAs in Ubuntu
  - sysctl net.ipv4.tcp_available_congestion_control

Output Example:

```
jemurray@home-server:~$ sysctl net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = reno cubic
```

TCP CCA in Linux (contd.)

- List installed, but not necessarily loaded CCAs
  - ls -al /lib/modules/`uname -r`/kernel/net/ipv4/tcp*

Output Example:

```
rver:~$ ls -al /lib/modules/`uname -r`/kernel/net/ipv4/tcp*
t root 14113 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_bbr.ko
t root 10913 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_bic.ko
t root 13865 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_cdg.ko
t root 10041 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_dctcp.ko
t root  8401 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_diag.ko
t root  7489 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_highspeed.ko
t root  9985 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_htcp.ko
t root  8073 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_hybla.ko
t root  8985 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_illinois.ko
t root  8617 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_lp.ko
t root 12561 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_nv.ko
t root 13441 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_probe.ko
t root  6449 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_scalable.ko
t root 11073 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_vegas.ko
t root  7609 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_veno.ko
t root  8097 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_westwood.ko
t root  7945 May  7 19:44 /lib/modules/4.15.0-144-generic/kernel/net/ipv4/tcp_yeah.ko
```

TCP CCA in Linux (contd.)

- Use modprobe to load additional modules
  - sudo /sbin/modprobe tcp_bbr

Output Example:

```
jemurray@home-server:~$ sudo /sbin/modprobe tcp_bbr

jemurray@home-server:~$ sysctl net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = reno cubic bbr
```

TCP CCA in Linux (contd.)

- Use sysctl to temporarily change the CCA to bbr
  - sudo /sbin/sysctl –w net.ipv4.tcp_congestion_control=bbr

```
jemurray@home-server:~$ sudo /sbin/sysctl -w net.ipv4.tcp_congestion_control=bbr
net.ipv4.tcp_congestion_control = bbr

jemurray@home-server:~$ sysctl net.ipv4.tcp_congestion_control
net.ipv4.tcp_congestion_control = bbr
```

TCP Code in Ubuntu

- Tentative location: /net/ipv4/tcp.c
- A snapshot of a part of the code …

elixir.bootlin.com/linux/latest/source/net/ipv4/tcp.c

Apps | e | WHO | mhGAP Hu... | D | eRa | Publications - A. B.... | Dear | ear | mangOH™ - open s... | 5(a) [Defining end-... | Online Application f... | ear

linux

Filter tags

- v6
  - v6.2
  - v6.1
    - v6.1.11
    - v6.1.10
    - v6.1.9
    - v6.1.8
    - v6.1.7
    - v6.1.6
    - v6.1.5
    - v6.1.4
    - v6.1.3
    - v6.1.2
    - v6.1.1
    - v6.1
    - v6.1-rc8
    - v6.1-rc7
    - v6.1-rc6
    - v6.1-rc5
    - v6.1-rc4
    - v6.1-rc3
    - v6.1-rc2

/ net / ipv4 / tcp.c

All symbols | Search Identifier

```
2734                    tcp_call_bpf_2arg(sk, BPF_SOCK_OPS_STATE_CB, oldstate, state);
2735
2736            switch (state) {
2737            case TCP_ESTABLISHED:
2738                    if (oldstate != TCP_ESTABLISHED)
2739                            TCP_INC_STATS(sock_net(sk), TCP_MIB_CURRESTAB);
2740                    break;
2741
2742            case TCP_CLOSE:
2743                    if (oldstate == TCP_CLOSE_WAIT || oldstate == TCP_ESTABLISHED)
2744                            TCP_INC_STATS(sock_net(sk), TCP_MIB_ESTABRESETS);
2745
2746                    sk->sk_prot->unhash(sk);
2747                    if (inet_csk(sk)->icsk_bind_hash &&
2748                        !(sk->sk_userlocks & SOCK_BINDPORT_LOCK))
2749                            inet_put_port(sk);
2750                    fallthrough;
2751            default:
2752                    if (oldstate == TCP_ESTABLISHED)
2753                            TCP_DEC_STATS(sock_net(sk), TCP_MIB_CURRESTAB);
2754            }
2755
2756            /* Change state AFTER socket is unhashed to avoid closed
2757             * socket sitting in hash tables.
2758             */
2759            inet_sk_state_store(sk, state);
2760    }
2761    EXPORT_SYMBOL_GPL(tcp_set_state);
2762
```

linux v6.1.11 powered by Elixir 2.1

An Advanced Command

- traceroute
  - Determines the route to a node
    - tracert: Windows
    - traceroute: Unix

A Sample Output of “tracert”

C:\Windows\system32\cmd.exe

```
C:\Users\User>tracert google.com

Tracing route to google.com [142.251.10.102]
over a maximum of 30 hops:

  1     3 ms     2 ms     1 ms  192.168.0.1
  2    10 ms     4 ms     2 ms  10.0.2.1
  3     5 ms    13 ms     2 ms  10.100.0.1
  4    44 ms    30 ms     3 ms  43.224.112.136
  5     4 ms     5 ms     6 ms  103.124.224.10
  6    52 ms    60 ms    50 ms  103.124.224.140
  7     *        *        *     Request timed out.
  8     *        *       51 ms  209.85.243.27
  9   635 ms   873 ms   120 ms  108.170.240.172
 10     *        *        *     Request timed out.
 11    54 ms    55 ms     *     216.239.41.225
 12    77 ms     *        *     142.251.52.67
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
 24    51 ms    58 ms    53 ms  sd-in-f102.1e100.net [142.251.10.102]

Trace complete.

C:\Users\User>
```

How Does Traceroute Work?

1. Launch a probe packet towards DST, with a TTL of 1
2. Every router hop decrements the IP TTL of the packet by 1
3. When the TTL hits 0, packet is dropped, router sends ICMP TTL Exceed packet to SRC with the original probe packet as payload
4. SRC receives this ICMP message, displays a traceroute "hop"
5. Repeat from step 1, with TTL incremented by 1 each time, until…
6. DST host receives probe, returns ICMP Dest Unreachable

Responses in Traceroute

- All hops except the last one will (or should) return a "TTL exceeded in transit" message
- The last hop should return a "destination unreachable/port unreachable" message indicating that it cannot handle the received traffic
  - UDP traceroute packets are typically addressed to a pseudorandom high port on which the end host is not likely to be listening

Traceroute Report Hop

- Traceroute packet with TTL of 1 enters router via the ingress interface
- Router decrements TTL to 0, drops packet, generates ICMP TTL Exceed
  - ICMP packet dst address is set to the original traceroute probe source (SRC)
  - ICMP packet src address is set to the IP of the ingress router interface
  - Traceroute shows a result based on the src address of the ICMP packet
  - The above traceroute will read: 172.16.2.1 10.3.2.2
  - You have NO visibility into the return path or the egress interface used

Traceroute Latency Calculation

- How is traceroute latency calculated?
  - Timestamp when the probe packet is launched
  - Timestamp when the ICMP response is received
  - Calculate the difference to determine round-trip time
  - Routers along the path do not do anytime "processing"
    - They simply reflect the original packet's data back to the SRC
    - Many implementations encode the original launch timestamp into the probe packet, to increase accuracy and reduce state
  - Most Importantly: only the ROUNDTRIP is measured
    - Traceroute is showing you the hops on the forward path
    - But showing you latency based on the forward **PLUS** reverse path
      - Any delays on the reverse path will affect your results

An Advanced Tool

- Nmap
  - Source: https://nmap.org/download.html#windows
  - Extremely powerful
  - Extremely invasive
  - Extremely obvious if you are not careful
  - Extremely illegal if not done correctly

Port Scanning Basics

- nmap scans more than 1660 ports
- Most port scanners list ports as opened or closed
- nmap recognizes 6 port states
  - Open
    - Accepting TCP connections or UDP packets
  - Closed
    - Host is up on the IP address
    - Accessible but no app is listening
    - Try later

Port Scanning Basics (contd.)

- nmap recognizes 6 port states (contd.)
  - Filtered
    - No response from probe
      - Firewall probably did a stealth drop
    - Forces nmap to retry many times
  - Unfiltered
    - Port is accessible but cannot determine whether open or closed
  - Open|Filtered
    - When unable to determine whether port is open of filtered
  - Closed|Filtered
    - When unable to determine whether port is closed or filtered

An Intense Scan in nmap: “Nmap Output”

Zenmap

Scan Tools Profile Help

Target: google.com Profile: Intense scan Scan Cancel

Command: nmap -T4 -A -v google.com

Hosts Services

OS Host

google.com (142.25

Filter Hosts

Nmap Output Ports / Hosts Topology Host Details Scans

nmap -T4 -A -v google.com Details

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-11 01:16 Bangladesh Standard Time
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 01:16
Completed NSE at 01:16, 0.00s elapsed
Initiating NSE at 01:16
Completed NSE at 01:16, 0.00s elapsed
Initiating NSE at 01:16
Completed NSE at 01:16, 0.00s elapsed
Initiating Ping Scan at 01:16
Scanning google.com (142.251.10.100) [4 ports]
NSOCK ERROR [0.4810s] ssl_init_helper(): OpenSSL legacy provider failed to load.

Completed Ping Scan at 01:16, 2.42s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:16
Completed Parallel DNS resolution of 1 host. at 01:16, 0.05s elapsed
Initiating SYN Stealth Scan at 01:16
Scanning google.com (142.251.10.100) [1000 ports]
Discovered open port 443/tcp on 142.251.10.100
Discovered open port 80/tcp on 142.251.10.100
Completed SYN Stealth Scan at 01:16, 5.41s elapsed (1000 total ports)
Initiating Service scan at 01:16
Scanning 2 services on google.com (142.251.10.100)
Service scan Timing: About 50.00% done; ETC: 01:18 (0:01:01 remaining)
Completed Service scan at 01:17, 71.30s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against google.com (142.251.10.100)
Retrying OS detection (try #2) against google.com (142.251.10.100)
Initiating Traceroute at 01:17
Completed Traceroute at 01:17, 4.08s elapsed
Initiating Parallel DNS resolution of 11 hosts. at 01:17
Completed Parallel DNS resolution of 11 hosts. at 01:17, 0.09s elapsed
NSE: Script scanning 142.251.10.100.
Initiating NSE at 01:17
Completed NSE at 01:17, 5.28s elapsed
Initiating NSE at 01:17
Completed NSE at 01:17, 6.61s elapsed
Initiating NSE at 01:17
Completed NSE at 01:17, 0.00s elapsed
Nmap scan report for google.com (142.251.10.100)
```

An Intense Scan in nmap : “Nmap Output” (contd.)

Zenmap

Scan Tools Profile Help

Target: google.com Profile: Intense scan Scan Cancel

Command: nmap -T4 -A -v google.com

Hosts Services

OS ◂ Host

google.com (142.25

Filter Hosts

Nmap Output Ports / Hosts Topology Host Details Scans

nmap -T4 -A -v google.com Details

```
Completed NSE at 01:17, 0.00s elapsed
Nmap scan report for google.com (142.251.10.100)
Host is up (0.032s latency).
Other addresses for google.com (not scanned): 142.251.10.113 142.251.10.102 142.251.10.138 142.251.10.101 142.251.10.139
rDNS record for 142.251.10.100: sd-in-f100.1e100.net
Not shown: 991 filtered tcp ports (no-response), 7 filtered tcp ports (admin-prohibited)
PORT     STATE SERVICE    VERSION
80/tcp   open  http       gws
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Date: Fri, 10 Feb 2023 19:16:20 GMT
|     Expires: -1
|     Cache-Control: private, max-age=0
|     Content-Type: text/html; charset=ISO-8859-1
|     P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
|     Server: gws
|     X-XSS-Protection: 0
|     X-Frame-Options: SAMEORIGIN
|     Set-Cookie: 1P_JAR=2023-02-10-19; expires=Sun, 12-Mar-2023 19:16:20 GMT; path=/; domain=.google.com; Secure
|     Set-Cookie: AEC=ARSKqsLqxFhbITg7HhuJkvnu0NVZrz6Notwo-iyl46eDZ-nwpxbIPLZdGg; expires=Wed, 09-Aug-2023 19:16:20 GMT; path=/; domain=.google.com; Secure;
HttpOnly; SameSite=lax
|     Set-Cookie: NID=511=s4xxb9RJs1426Kb4pxEK0KgMtHwaBW09uwe88t8IOYwgSBRS3oJD6ZwFHThCukTWVV0GxLs6nUaUf_K7410gdl2zFXs7LuhFX3Rk-
HaKiEsTXfRvBDYqnXtm059jzU_wJtQDDGtjPe3YwOylBm09bIHyFTnpCNEq3QmY0xjHjBw; expires=Sat, 12-Aug-2023 19:16:20 GMT; path=/; domain=.google.com; HttpOnly
|     Accept-Ranges: none
|     Vary: Accept-Encoding
|     <!doctype html><html
|   HTTPOptions:
|     HTTP/1.0 405 Method Not Allowed
|     Allow: GET, HEAD
|     Date: Fri, 10 Feb 2023 19:16:21 GMT
|     Content-Type: text/html; charset=UTF-8
|     Server: gws
|     Content-Length: 1592
|     X-XSS-Protection: 0
|     X-Frame-Options: SAMEORIGIN
|     <!DOCTYPE html>
|     <html lang=en>
|     <meta charset=utf-8>
```

An Intense Scan in nmap : “Nmap Output” (contd.)

Zenmap

Scan Tools Profile Help

Target: google.com Profile: Intense scan Scan Cancel

Command: nmap -T4 -A -v google.com

Hosts Services

OS Host

google.com (142.25

Nmap Output Ports / Hosts Topology Host Details Scans

nmap -T4 -A -v google.com Details

```
Device type: general purpose
Running (JUST GUESSING): FreeBSD 7.X (88%)
OS CPE: cpe:/o:freebsd:freebsd:7.0
Aggressive OS guesses: FreeBSD 7.0-STABLE (88%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.000 days (since Sat Feb 11 01:17:31 2023)
Network Distance: 22 hops
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   8.00 ms  192.168.0.1
2   8.00 ms  10.0.2.1
3   8.00 ms  10.100.0.1
4   8.00 ms  43.224.112.136
5   5.00 ms  103.124.224.10
6   58.00 ms 103.124.224.140
7   96.00 ms 72.14.205.188
8   49.00 ms 209.85.243.57
9   53.00 ms 108.170.240.164
10  53.00 ms 142.251.230.215
11  ...
12  51.00 ms 142.251.51.213
13  ... 21
22  52.00 ms sd-in-f100.1e100.net (142.251.10.100)

NSE: Script Post-scanning.
Initiating NSE at 01:17
Completed NSE at 01:17, 0.00s elapsed
Initiating NSE at 01:17
Completed NSE at 01:17, 0.00s elapsed
Initiating NSE at 01:17
Completed NSE at 01:17, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 100.89 seconds
           Raw packets sent: 2127 (98.226KB) | Rcvd: 85 (4.634KB)
```

Filter Hosts

An Intense Scan in nmap: “Ports/Hosts”

Zenmap

Scan Tools Profile Help

Target: google.com Profile: Intense scan Scan Cancel

Command: nmap -T4 -A -v google.com

Hosts Services

OS Host

google.com (142.25

Filter Hosts

Nmap Output Ports / Hosts Topology Host Details Scans

Port	Protocol	State	Service	Version
80	tcp	open	http	gws
443	tcp	open	https	gws

An Intense Scan in nmap: “Topology”

An Intense Scan in nmap: “Host Details”

Zenmap

Scan Tools Profile Help

Target: google.com Profile: Intense scan Scan Cancel

Command: nmap -T4 -A -v google.com

Hosts Services

OS Host

google.com (142.25

Filter Hosts

Nmap Output | Ports / Hosts | Topology | Host Details | Scans

google.com (142.251.10.100)

Host Status

State: up

Open ports: 2

Filtered ports: 998

Closed ports: 0

Scanned ports: 1000

Up time: 18

Last boot: Sat Feb 11 01:17:31 2023

Addresses

IPv4: 142.251.10.100

IPv6: Not available

MAC: Not available

Hostnames

Name - Type: google.com - user

Name - Type: sd-in-f100.1e100.net - PTR

Operating System

Name: FreeBSD 7.0-STABLE

Accuracy: 88%

Ports used

OS Classes

TCP Sequence

IP ID Sequence

TCP TS Sequence

A Regular Scan in nmap: “Nmap Output”

Thank You

Advanced Topics - 2

(Ack: Ishrat Jahan)

Load Balancing

Load Balancing

High Availability

VRRP - Virtual Router Redundancy Protocol

SDN (Software Defined Networking)

(a) Traditional networking

CP - Control Plane
DP - Data Plane

(b) SDN

Network Virtualization

Server Virtualization

Network Virtualization (Overlay vs Underlay)

VXLAN

For details, please check [this]() website.

Data Centers and Disaster Recovery

DC-DR

Cold site
Little or no equipment
No network connectivity
Not ready for automatic failover
No data synchronization
High risk of data loss
Cheap
Warm site
Partially redundant equipment
Network connectivity is enabled
Failover occ within hou
Daily eekly data synchronization
Minimum data loss
Cost-effective
Hot site
Fully redundant equipment
Network connectivity is enabled
Failover occurs within hours or days
Near real-time data synchronization
Zero data loss
Expensive

Earthquake Zones – An Important Consideration for DR

Thank You!

NS-3 Basics

CSE 322 - Computer Networking Sessional

Special Credits:

1. Syed Md Mukit Rashid, Assistant Professor (on leave), CSE, BUET. PhD Student @ Pennsylvania State University.
2. Mashiat Mustaq, Lecturer (retd), CSE, BUET. PhD Student @ Purdue University.

NS3 installation steps

- Version to be used - **3.43**
- *https://www.nsnam.org/docs/release/3.43/installation/html/quick-start.html*.
- You may check out this for a video walkthrough.
- Follow the exact steps mentioned in the link. Install the **prerequisites** first.
- After following the steps, run the following command :

```
$ ./ns3 run hello-simulator
```

- If it outputs "`Hello Simulator`", then it was installed correctly.

NS3 installation steps

- *Some modules may not build due to missing dependency,* which won't be a problem. You may solve this error by installing the missing dependencies if you wish.

- Python bindings are not required for this course.

NS3 - A Network Simulator

- **Resources:**
  - **Official Website :** https://www.nsnam.org/
  - **Tutorial** : https://www.nsnam.org/docs/release/3.43/tutorial/ns-3-tutorial.pdf
    - Useful chapters : 5-9
  - **Models:** https://www.nsnam.org/docs/release/3.43/models/html/index.html
    - Description of models are provided here. Helpful for understanding the concepts.
  - **Doxygen API documentation:**
    - https://www.nsnam.org/docs/release/3.43/doxygen/index.html
  - **IDEs :** VSCode, Jetbrains CLion etc (Install the necessary plugins)
  - **Google group :** https://groups.google.com/g/ns-3-users
  - **YouTube Playlist :** https://youtu.be/bjUNbXBmA2c?feature=shared

Conceptual Overview - Node

- A basic computing device abstraction, e.g : Computer
- A class defined in C++
- Purpose :
  - Adding functionality such as applications, protocol stacks, peripheral cards etc

Conceptual Overview - Application

- Representation of user-level software applications
- A class defined in C++
- Purpose :
  - Runs on nodes to to run different types of simulations
- Examples:
  - `UDPEchoServer/ClientApplication`, `BulkSendApplication`, `OnOffApplication`, `PacketSinkApplication`etc
  - built- in applications directory : `src/application/models`

Conceptual Overview - Channel

- Abstraction of the media through which data flows in the network
- Connect Node to a channel
- A class defined in C++
- Types :
  - `CsmaChannel` (Ethernet)
  - `PointToPointChannel`
  - `WifiChannel`
  - `WimaxChannel` etc

Conceptual Overview - Net Device

- Abstraction of both software driver and Network Interface Card used to connect a Node to a network
- A net device is “installed” in a *Node* to enable the *Node* to communicate with other Nodes via *Channels*.
- A *Node* may be connected to more than one *Channel* via multiple *NetDevices*.
- Types :
  - `CsmaNetDevice` (Ethernet)
  - `PointToPointNetDevice`
  - `WifiNetDevice`

Conceptual Overview - Topology Helpers

In each simulation, simplify common tasks such as:

- Connecting Nodes to NetDevices
- Connecting NetDevices to Channels
- Assigning IP addresses etc.

combine those many distinct operations into an easy to use model for your convenience

Examples: `PointToPointHelper`, `InternetStackHelper`, `UDPEchoServerHelper` etc.

Simulation Overview

Run first.cc

```
cp examples/tutorial/first.cc
scratch/
```

```
./ns3 build
```

```
./ns3 run scratch/first
```

When the simulator will stop?

- No further events are in the event queue.
- A special Stop event is found.
  - `Simulator::Stop(stopTime)`
  - Necessary when there are recurring events (WiFi)
  - *Important to call* `Simulator::Stop` *before calling* `Simulator::Run`

Logging Overview

Log Type	Purpose	Macro
LOG_ERROR	error messages	NS_LOG_ERROR
LOG_WARN	warning messages	NS_LOG_WARN
LOG_DEBUG	relatively rare, ad-hoc debugging messages	NS_LOG_DEBUG
LOG_INFO	informational messages about program progress	NS_LOG_INFO
LOG_FUNCTION	a message describing each function called	NS_LOG_FUNCTION - member func. NS_LOG_FUNCTION_NOARGS - static func.
LOG_LOGIC	messages describing logical flow within a function	NS_LOG_LOGIC
LOG_ALL	Log everything mentioned above	no associated macro

Logging Overview

- **LOG_LEVEL_TYPE:** Enables logging of all the levels above it.
  - **Ex : LOG_LEVEL_INFO :** Enable logging for ERROR, WARN, DEBUG, INFO types.
- **NS_LOG_UNCOND** – Log the associated message unconditionally (no associated log level).

Logging Overview

- Using the shell environment variable -> NS_LOG
  - increase the logging level without changing the script
    - `export NS_LOG=UdpEchoClientApplication=level_all`
  - Enable two logging components together - Colon separated
    - `export`

      `'NS_LOG=UdpEchoClientApplication=level_all|prefix_func:UdpEchoServerApplication=level_all'`

Logging Overview

- Using the shell environment variable -> NS_LOG
  - Distinguish which method generates a log message - ORing
    - `export`

      ```
      'NS_LOG=UdpEchoClientApplication=level_all|prefix_func'
      ```
  - See the simulation time
    - `export`

      ```
      'NS_LOG=UdpEchoClientApplication=level_all|prefix_time'
      ```

Logging Overview

Prefix Symbol	Meaning
LOG_PREFIX_FUNC	Prefix the name of the calling function.
LOG_PREFIX_TIME	Prefix the simulation time.
LOG_PREFIX_NODE	Prefix the node id.
LOG_PREFIX_LEVEL	Prefix the severity level.
LOG_PREFIX_ALL	Enable all prefixes.

Logging Overview

NS_LOG Wildcards

The log component wildcard `*' will enable all components. To enable all components at a specific severity level use `*=<severity>`.

The severity level option wildcard `*' is a synonym for `all`. This must occur before any `|' characters separating options. To enable all severity classes, use `<log-component>=*`, or `<log-component>=*|<options>`.

The option wildcard `*' or token `all` enables all prefix options, but must occur *after* a `|' character. To enable a specific severity class or level, and all prefixes, use `<log-component>=<severity>|*`.

The combined option wildcard `**` enables all severities and all prefixes; for example, `<log-component>=**`.

The uber-wildcard `***` enables all severities and all prefixes for all log components. These are all equivalent:

```
$ NS_LOG="***" ...          $ NS_LOG="*=all|*" ...          $ NS_LOG="*=*|all" ...
$ NS_LOG="*=**" ...         $ NS_LOG="*=level_all|*" ...    $ NS_LOG="*=*|prefix_all" ...
$ NS_LOG="*=*|*" ...
```

Logging Overview

- Turn off logging previously enabled
  - `export NS_LOG=""`

- Enable logging in code
  - `export NS_LOG=FirstScriptExample=info`

Using Command Line Arguments

- Declare command line parser.

- Show general arguments for a program.
  - ```
    ./ns3 run "scratch/first --PrintHelp"
    ```

- Provide new command line argument
  - ```
    ./ns3 run "scratch/first
    --ns3::PointToPointNetDevice::DataRate=32Kbps"
    ```

- Provide multiple command line arguments
  - ```
    ./ns3 run "scratch/first
    --ns3::PointToPointNetDevice::DataRate=32Kbps
    --ns3::PointToPointChannel::Delay=2ms"
    ```
    ```

Using User Defined Command Line Arguments

```
int
main(int argc, char *argv[])
{
uint32_t nPackets = 1;
CommandLine cmd;
cmd.AddValue("nPackets", "Number of packets to echo", nPackets);
cmd.Parse(argc, argv);
… … …
```

ASCII Tracing

+	An enqueue operation occurred on the device queue
-	A dequeue operation occurred on the device queue
d	A packet was dropped, typically because the queue was full
r	A packet was received by the netdevice

ASCII Tracing

+	Enqueue
2	Time (Seconds)
/NodeList/0/DeviceList/0/$ns3::PointToPointNetDevice/TxQueue/Enqueue	Trace source origin
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0 DSCP Default ECN Not-ECT ttl 64 id 0 protocol 17 offset (bytes) 0 flags [none] length: 1052 10.1.1.1 > 10.1.1.2) ns3::UdpHeader (length: 1032 49153 > 9) Payload (size=1024)	Packet information

Pcap Tracing

- Wireshark
- Tcpdump
  - tcpdump -nn -tt -r filename.pcap

The command `tcpdump -nn -tt -r <filename>` is used to read and analyze packet data from a previously captured file. Here's a breakdown of the options:

- `-nn` : Stops `tcpdump` from resolving hostnames and service names, displaying IP addresses and port numbers in numeric form. This makes the output faster and easier to interpret without DNS or port name lookups.
- `-tt` : Prints timestamps in raw format as absolute time since the epoch (UNIX time in seconds), which is useful for analyzing timing without human-readable date formatting.
- `-r <filename>` : Reads packets from a capture file (`<filename>`) instead of live traffic. The file should be in the pcap format, commonly generated by `tcpdump` or other packet capture tools.

ChatGPT

Ethernet (Bus Network)

- CSMA NetDevice and channel
- Promiscuous mode - allows a network device to intercept and read each packet.
- ARP (Address Resolution Protocol) - retrieves the receiver’s MAC address.

Wireless Network

- AP - Access Point
- AP generates beacons continuously
- Beacon - regular transmissions from access points (APs)
    - purpose to inform user devices (clients) about available Wi-Fi services and nearby access points