Network Device Configuration

CS Capstone

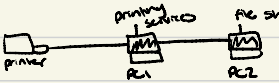
January Learning

Notes

NETWORKING BASICS

PEER TO PEER NETWORK

- hosts function as both clients and servers simultaneously



- easy to set up
- lower cost
- good for simple tasks
- not scalable
- less secure
- slower performance

NETWORK REPRESENTATIONS

- **Network interface cards (NIC)** = physically connects end device to network
- **interface** = specialized port that connects to an individual network
- Physical Topology = physical location of devices + cables
- Logical Topology = devices, ports, addressing schemes

Create internet standards:

- **Internet Engineering Task Force (IETF)**
- **Internet Corporation for Assigned Names + Numbers (ICANN)**
- Internet Architecture Board (IAB)

Options for internet connections

- cable = transmits on same line as TV, high bandwidth + availability, always on
- Digital Subscriber Line (DSL) = runs on phone line, HB + HA
- cellular = connects over cell phone network
- satellite = requires clear line of sight to satellite
- Dial-up = uses phone line + modem, low bandwidth

- Dedicated Leased Lines = reserved WAN lines connecting locations, rented out
- Metro Ethernet = extend LAN tech into WAN
- Business DSL = symmetric (upload = download same high speed)
- Satellite

BASIC SWITCH & END DEVICE CONFIGURATION

- Cisco uses Cisco Internetwork Operating System (Cisco IOS)

- Console : out of band, physical management port
  requires computer w/ terminal emulation software
- SSH : in-band, recommended
  requires active networking services on device
- Telnet : in-band, not encrypted, do not use

- PuTTY, TeraTerm, SecureCRT

primary command modes

- User EXEC mode: basic monitoring, cannot change config, > symbol
- privileged EXEC mode: change config, # symbol
  ↳ move between w/ enable | disable

- To configure first enter global config mode Switch(config)#
  then enter sub-config mode
  ↳ line config mode: config console, SSH access line console 0
  Switch(config-line)#
  ↳ interface config mode: config switch port / router interface interface fastEthernet 0/1
  Switch(config-if)#

  exit to go one step lower, end to go lower to priv. EXEC mode

command structure	· prompt command arguments Switch> show ip protocols · enter ? in CLI to see available commands/arguments at level
basic switch config	· conf t hostname <name> · Secure user EXEC mode: conf t line console 0 password <pass> login end priv EXEC: conft enable secret <pass> exit · enable secret <pass> overrides enable password <pass> · Secure Telnet/SSH access: conf t line vty 0 15 password <pass> login end · Encrypt plaintext passwords: (config) service password-encryption

	· Create banner: (config) banner motd #message#
Configuration files	· startup-config: saved, runs in NVRAM, runs on startup/reboot · running-config: not permanent, stored in RAM, current config, effect startup changes
	· >show running-config · #show startup-config
	· #copy running-config startup-config: makes rc changes permanent
	· #reload: reboots the device, warning on unsaved running-config · #erase startup-config: deletes su config. Then reload device to return to device defaults
save config to TXT	· enable show logging in PuTTY / Tera, enter file name · run show (config file) · disable logging

access switch remotely	(config)# interface vlan 1 (config-if)# ip address <ip> <subnet mask> (config-if)# no shutdown (config)# ip default-gateway <ip>
switch boot sequence	① switch runs power on self test (POST) program stored in ROM. checks CPU subsystem, DRAM, and flash device ② loads boot loader software stored in ROM ③ boot loader initializes CPU registers ④ BL initializes flash file system ⑤ BL loads default IOS software image : transfers control to OS · switch boots w/ file defined in boot variable. if not set uses first executable found · IOS initialized interfaces using startup config → config.text #show boot : displays name of current boot file ① connect switch to PC w/ console cable : terminal emulator ② Unplug switch power cord, reconnect, hold mode button while LED flashes ③ switch: prompt will appear

	: set shows path of BOOT env var : flash_init initializes flash file system : dir flash view directories : files : BOOT = flash:<file path> changes BOOT env var path : boot loads new IOS
switch verification commands	# show interfaces (id) # show startup-config # show running-config # show flash → info about flash file system # show history → command history # show ip interface # show mac-address-table
frame errors	• runts : frames < 64 bytes on ethernet, too small to transmit • giants : frames > 1,518 bytes on ethernet, too large to transmit • CRC : calculated checksum not same as received checksum • Output Errors : sum of all errors preventing final datagram transmission • Input Errors : total number of errors • Collisions : number of messages retransmitted because of collisions • Late Collisions : collisions after 512 bits transmitted

CONFIGURE SWITCH PORTS

	· full-duplex : both ends of connection send & receive simultaneously requires microsegmentation → port has only 1 connected device, no collision domain · half-duplex : one directional data flow. Problems w/ collisions
manually configure	· sometimes switches come configured to auto - they determine duplex depending on duplex speed of the connected device. Typically only full duplex if speed is 10000 Mbps / 10 Gb · best practice manually configure when connecting to known devices like servers, other network devices, dedicated PCs (config)# interface FastEthernet 0/1 (config-if)# duplex full # speed 1000 ← in mb/s # end
auto - MDIX	· typically end devices, routers connected to switch w/ straight through cable switches, repeaters connected w/ crossover cable · using automatic medium-dependent interface crossover detects otherwise vs. required cable : configures connection to work no matter cable type. (config-if)# mdix auto # ethernet-controller fa0/1 any	inclusive MDIX

CONFIGURE SECURE REMOTE ACCESS

- SSH on port 22
- use instead of Telnet
- encrypted connection

show ip ssh : verify switch supports ssh

(config) # ip domain-name <name>

(config) # ip ssh version 2

(config) # crypto key generate rsa : enables SSH server w/ key pair

(config) # username <> secret <> : configure authentication

(config) # line vty 0 15

(config-line) # transport input ssh ← enable ssh, disable other types

login local ← users must be from local database

exit

(config) # crypto key zeroize rsa : deletes keys, disables server

VLANs

- VLAN = group of end points w/ common set of requirements
  - ↳ independent of physical location
  - ↳ within same broadcast domain
  - ↳ need to be routed to communicate w/ other VLANs
  - ↳ layer 2 segment

- Access ports = belong to single VLAN, connects end devices
- ports can be assigned to VLANs
  - ① Statically — by admin
  - ② Dynamically — by VLAN policy server

VLAN Ranges

VLAN	Type	Description
0	Reserved	802.1P
1	Normal	Default native VLAN
2 - 1001	Normal	
1002 - 1005	Normal	Default token ring / FDDI VLANs
1006 - 4094	Extended	Extended Ethernet VLANs
4095	Reserved	System use

Configure VLAN

```
(config) # vlan <id>
(config-vlan) # name <name>
```
← create VLAN

```
(config) # interface fastethernet 0/2
(config-if) # switchport mode access
(config-if) # switchport access vlan <id>
```
← assign port to VLAN

- Trunk ports = carry data from multiple VLANs
  2 associated protocols:
  1. **Inter Switch Link (ISL)**
     Preserves source VLAN ID info for frames traversing trunk. Frame encapsulated w/ VLAN header that is removed at receiving switch.
     - ↳ uses per VLAN Spanning Tree to avoid loops
     - ↳ Frame has 3 main fields:
       1) ISL header, encapsulates OG frame
       2) The OG frame
       3) Frame Check Sequence (FCS) for error checking at end.
  2. **IEEE 802.1Q**
     Inserts 4 byte tag into OG frame between SA field & type/length fields for FCS error checking
     - ↳ has a User Priority field, 3 bit, levels 0-7
     - ↳ 12 bit VLAN ID field, 0-4095

configure trunk

```
(config) # int fast 0/2
(config-if) # switchport
        # switchport trunk encap dot1q/isl/negotiate
        # switchport mode trunk
```

- **Dynamic Trunking Protocol (DTP)**
  - Cisco protocol negotiates common trunking mode between 2 switches: 802.1Q or ISL
  - ↳ Dynamic Desirable = will try to become trunk
  - ↳ Dynamic Auto is passive, will only become trunk if other port is DD

configure dynamic trunking

```
(config)# int f0/11
(config-if)# switchport mode dynamic auto
```
} S1

```
(config)# int f0/11
(config-if)# switchport mode dynamic desirable
```
} S2

```
# show interfaces trunk
```
} check

filter VLANs

```
(config-if)# switchport trunk allowed vlan 1, 10, 20      ← allow specified
# switchport trunk allowed vlan except 10-30               ← allow all BUT
# switchport trunk allowed vlan remove 1, 10, 20           ← remove allowed
```

VTP

VLAN Trunking Protocol	· Layer 2 cisco messaging protocol · Manage creation, deletion, renaming of VLANs on switches in same domain · Allow VLAN info to spread through switched network automatically ↳ switches all have consistent VLAN info ↳ reduces manual admin overhead
VTP Domains	· Group of connected switches in same VTP management domain · Switch can only be in one domain · Switch will drop VTP packets from other domains
VTP modes	**Server mode**: · Default mode ← "administrative switch" · Controls creation, deletion, modification of VLANs in domain · Store VLAN database in NVRAM, advertise to other switches in domain · Each domain needs at LEAST one switch in server mode **Client mode**: · Advertises : receives VTP info · DOES NOT allow VLAN modification or store database in NVRAM **Transparent Mode**: · Ignore VTP updates · Allow VLAN modification ↳ ONLY LOCALLY, not propagated to others

VTP Advertisements	· Sent out through trunk link · Configuration Revision number determines which info is most recent ↳ highest # wins ↳ always ensure new switches start w/ low # (config)# vtp mode <mode> (config)# vtp domain <domain name> ← add to domain # show vtp status ← show info
VTP Pruning	· increase bandwidth by restricting flooded traffic to trunk lines that must be used to get to receiver only · Disabled by default # vtp pruning ← enable (only for server mode)

VOICE : WIRELESS VLANS

Voice

Voice : data

- link between IP phones : switch need to be trunk in order to separate
- just need config on switch port

```
(config)# int fast 0/1
(config-if)# switchport voice vlan <vlan option>
```

1. None
   - No trunk
   - Access VLAN for both
2. ID # (if insane)
   - Trunk
   - Voice on VLAN #
   - Data untagged
3. dot1p
   - Trunk
   - Voice VLAN 0
   - Data untagged
4. untagged
   - Trunk
   - All untagged

Wireless

```
(config)# int <interface ID>
(config-if)# switchport trunk encapsulation dot1q
# switchport trunk allowed vlan <wireless VLANs>
# switchport mode trunk
```

PROTOCOL SUITES

TCP/IP Protocol Suite

- developed by Internet Engineering Task Force (IETF)

Suite Layer	Protocol Stack
Application	HTTP DNS DHCP/SLAAC SMTP/IMAP PTP
Transport	TCP UDP
Internet	IP ICMP Routing Protocols
Network	Ethernet/WLAN ARP

- develop standards used to create communication standards
- Institute of Electrical & Electronics Engineers (IEEE): create power, energy, networking standards like 802.3 & 802.11
- Electronic Industries Alliance (EIA): wiring, connectors, racks standards
- Telecommunications Industry Association (TIA): radio, cell tower, satellite standards
- International Telecommunications Union-Telecommunication Standardization Sector (ITU-T)
  Video compression, DSL, broadband standards

OSI Model	Application - process to process communications Presentation - common representation of data as it is transferred between applications Session - manage data exchange Transport - segments, transfers, reassembles data between end devices Network - exchange data packs over network Data Link - exchange data frames over a common medium Physical - actual means to activate connections for a bit transmissions
protocol data units	Data → Segment → Packet → Frame → Bits
	• Network Layer Source + Destination Addresses: deliver IP packet • Data Link Layer Source + Destination Addresses: deliver frame between NICs

BASIC ROUTER CONFIGURATION

Basic Router Config * when we switch *	(config) # hostname <name> → name router (config) # enable secret <password> → secure privileged EXEC (config-line) # password <password> — secure user EXEC # login (config-line) # line vty 0 4 — secure remote access # password <password> # login # transport input { ssh | telnet } (config) # service password-encryption → encrypt passwords in config # banner motd "message" → legal banner () # copy run start → save config to NVRAM
configure router interfaces	(config) # interface <type> <number> ← ex) gigabitEthernet 0/0/0 (config-if) # description <text> # ip address <address> <mask> # no shutdown ← activates interface
	# show ip interface brief : show all interfaces, ip addresses, their status # show ip route : show IP routing tables # show interfaces : statistics for all interfaces on device

default gateway

- one router on network must be default gateway

on a switch

- (config)# ip default-gateway <ip address>

Device	Interface	IP	Mask	D.G
R1	G 0/0	192.168.10.1	/24	—
	G 0/1	192.168.11.1	/24	—
S1	VLAN 1	192.168.10.2	124	G 0/0
S2	VLAN 1	192.168.11.2	/24	G 0/1
PC1		192.168.10.10	/24	G 0
PC2		192.168.10.11	/24	G 0
PC3		192.168.11.10	/24	G 1
PC4		192.168.11.11	/24	G 1

Test	Success?	Issue	Solution
PC1 to PC2	No	IP on PC1	Change PC1 IP
PC1 to S1	Yes		
PC1 to R1	Yes		
PC3 to PC4			
PC3 to S2			
PC3 to R1			