Securing the Campus Infrastructure

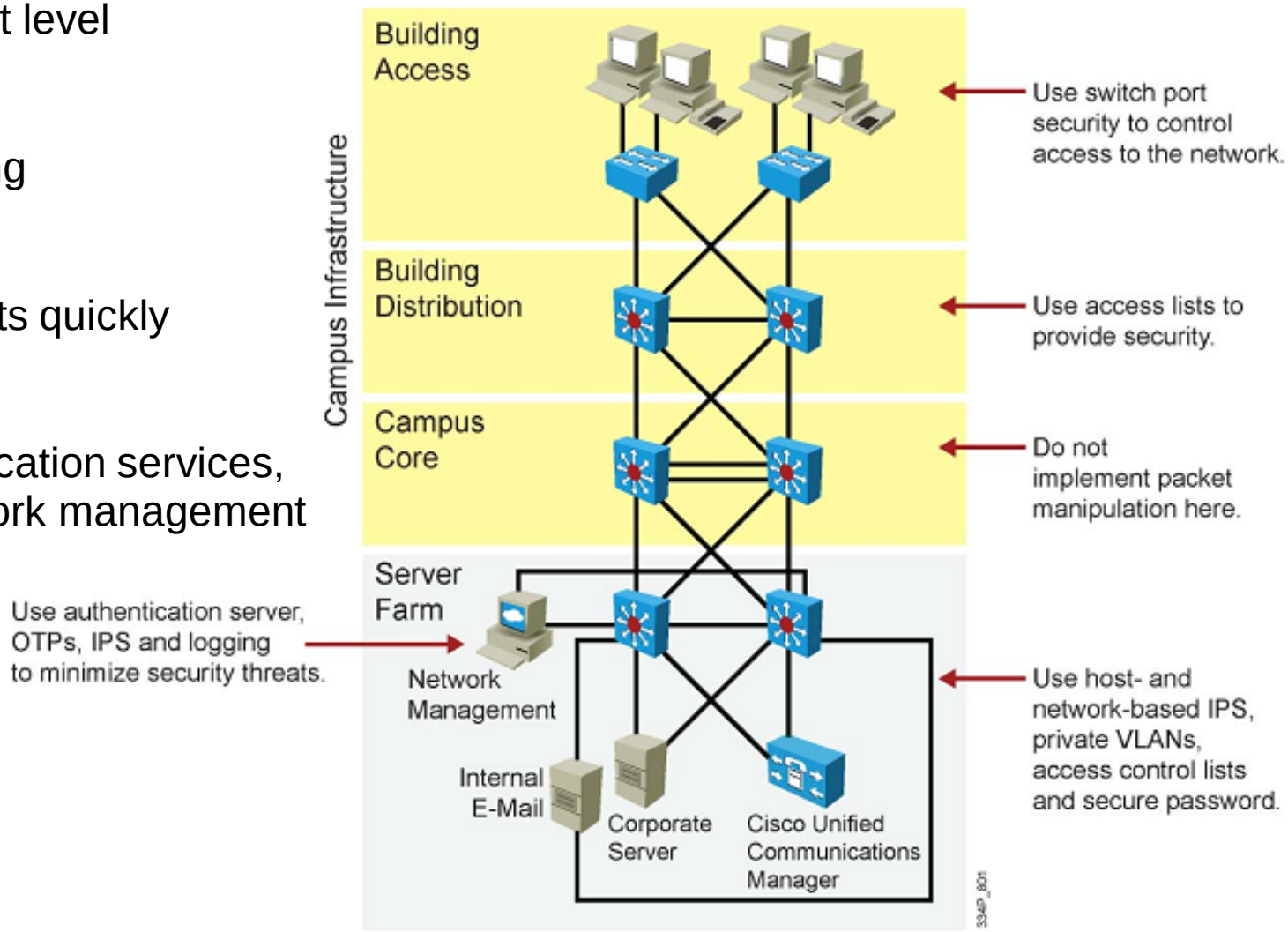
SWITCH Module 6

Agenda

- **Security Fundamentals**
- **MAC Layer Attacks**
  - CAM Overflow
  - Port Stealing
  - Port Security
- **VLAN Attacks and Switch Spoofing**
  - VTP exploit
  - Double-tagging VLAN hopping
  - Switch-spoofing
- **Identity Spoofing Attacks**
  - DHCP Spoofing
  - DHCP Snooping
  - ARP Spoofing
  - Dynamic ARP Inspection
  - IP Address Spoofing
  - IP Source Guard
- **Securing Remote Access**
  - Telnet
  - SSH
  - VTY ACLs
  - HTTPS
- **AAA**
  - Authentication
  - Authorization
  - Accounting
  - Radius + TACACS
  - 802.1X
- **Best Practices for Switches**

Security Infrastructure Services

- Access
  - control at port level
- Distribution
  - packet filtering
- Core
  - switch packets quickly
- Server farm
  - provide application services, include network management system.



Layer 2 Attack Categories ①

Attack Method	Description	Steps to Mitigation
MAC Layer Attacks		
MAC Address Flooding	Frames with unique, invalid source MAC addresses flood the switch, exhausting content addressable memory (CAM) table space, disallowing new entries from valid hosts. Traffic to valid hosts is subsequently flooded out all ports.	Port security. MAC address VLAN access maps.
VLAN Attacks		
VLAN Hopping	By altering the VLAN ID on packets encapsulated for trunking, an attacking device can send or receive packets on various VLANs, bypassing Layer 3 security measures.	Tighten up trunk configurations and the negotiation state of unused ports. Place unused ports in a common VLAN.
Attacks between Devices on a Common VLAN	Devices might need protection from one another, even though they are on a common VLAN. This is especially true on service-provider segments that support devices from multiple customers.	Implement private VLANs (PVLAN).

Layer 2 Attack Categories ②

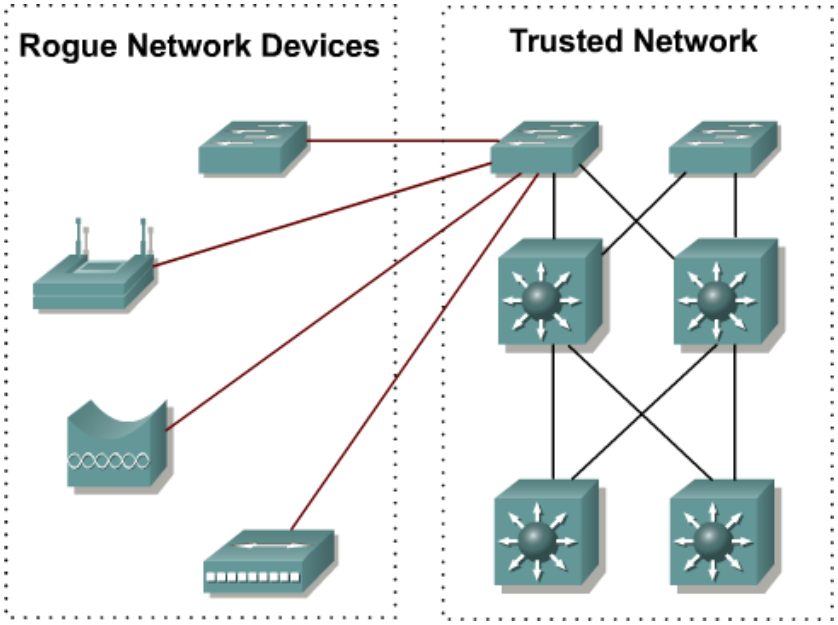
Attack Method	Description	Steps to Mitigation
Spoofing Attacks		
DHCP Starvation and DHCP Spoofing	An attacking device can exhaust the address space available to the DHCP servers for a period of time or establish itself as a DHCP server in man-in-the-middle attacks.	Use DHCP snooping.
Spanning-tree Compromises	Attacking device spoofs the root bridge in the STP topology. If successful, the network attacker can see a variety of frames.	Proactively configure the primary and backup root devices. Enable root guard.
MAC Spoofing	Attacking device spoofs the MAC address of a valid host currently in the CAM table. The switch then forwards frames destined for the valid host to the attacking device.	Use DHCP snooping, port security.
Address Resolution Protocol (ARP) Spoofing	Attacking device crafts ARP replies intended for valid hosts. The attacking device's MAC address then becomes the destination address found in the Layer 2 frames sent by the valid network device.	Use Dynamic ARP Inspection (DAI), DHCP snooping, port security.

Layer 2 Attack Categories ③

Attack Method	Description	Steps to Mitigation
Switch Device Attacks		
Cisco Discovery Protocol (CDP) Manipulation	Information sent through CDP is transmitted in clear text and unauthenticated, allowing it to be captured and divulge network topology information.	Disable CDP on all ports where it is not intentionally used.
Secure Shell Protocol (SSH) and Telnet Attacks	Telnet packets can be read in clear text. SSH is an option but has security issues in version 1.	Use SSH version 2. Use Telnet with vty ACLs.

Control Access Layer Enrollment

- Unwanted (and unfortunately unwanted) phenomenon are users plugging rogue devices into enterprise network
- Role of access layer switch is to control, secure and limit the access to network
- Several mechanisms are available
  - Port Security
  - 802.1X
  - Network Admission Control



Organization Security Policies

- You should consider the policies of an organization when determining what level of security and what type of security should be implemented
- A well-established security policy has the characteristics listed above
  - Provides a process for auditing existing network security
  - Provides a general security framework for implementing network security
  - Defines disallowed behaviors toward electronic data
  - Determines which tools and procedures are needed for the organization
  - Communicates consensus among a group of key decision makers and defines responsibilities of users and administrators
  - Defines a process for handling network security incidents
  - Enables an enterprise-wide, all-site security implementation and enforcement plan

Securing Switch Devices and Protocols

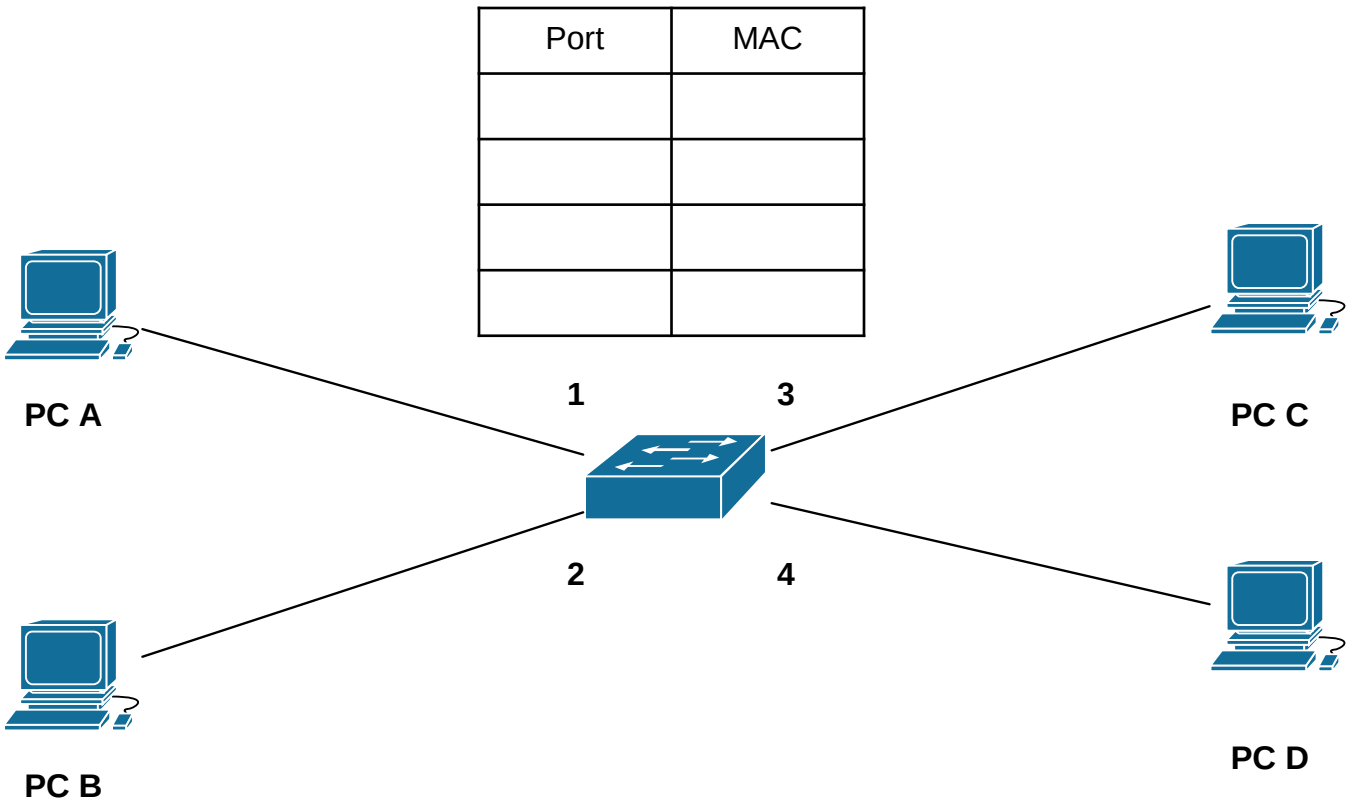
- Configure strong system passwords (**`enable secret`**)
- Restrict management access using ACLs (**`deny ip any any log`**)
- Secure physical access to the console
  - Console access requires a minimum level of security both physically and logically
- Secure access to vty lines
- Configure system warning banners
  - Clearly stating the ownership, usage, access, and protection policies prior to a login aids in stronger prosecution if unauthorized access occurs
- Disable unneeded or unused services
- Trim and minimize the use of CDP/LLDP (**`no [cdp | lldp] enable`**)
- Disable the integrated HTTP daemon where appropriate) (**`no ip http server`**)
- Configure basic system logging (syslog)
- Secure SNMP (use SNMPv3 with authentication and privacy)
- Limit trunking connections and propagated VLANs (**`switchport nonegotiate`**)
- Secure the spanning-tree topology (BPDU/Loop/Root Guard, BPDU Filter)

Disabling Unneeded or Unused Services

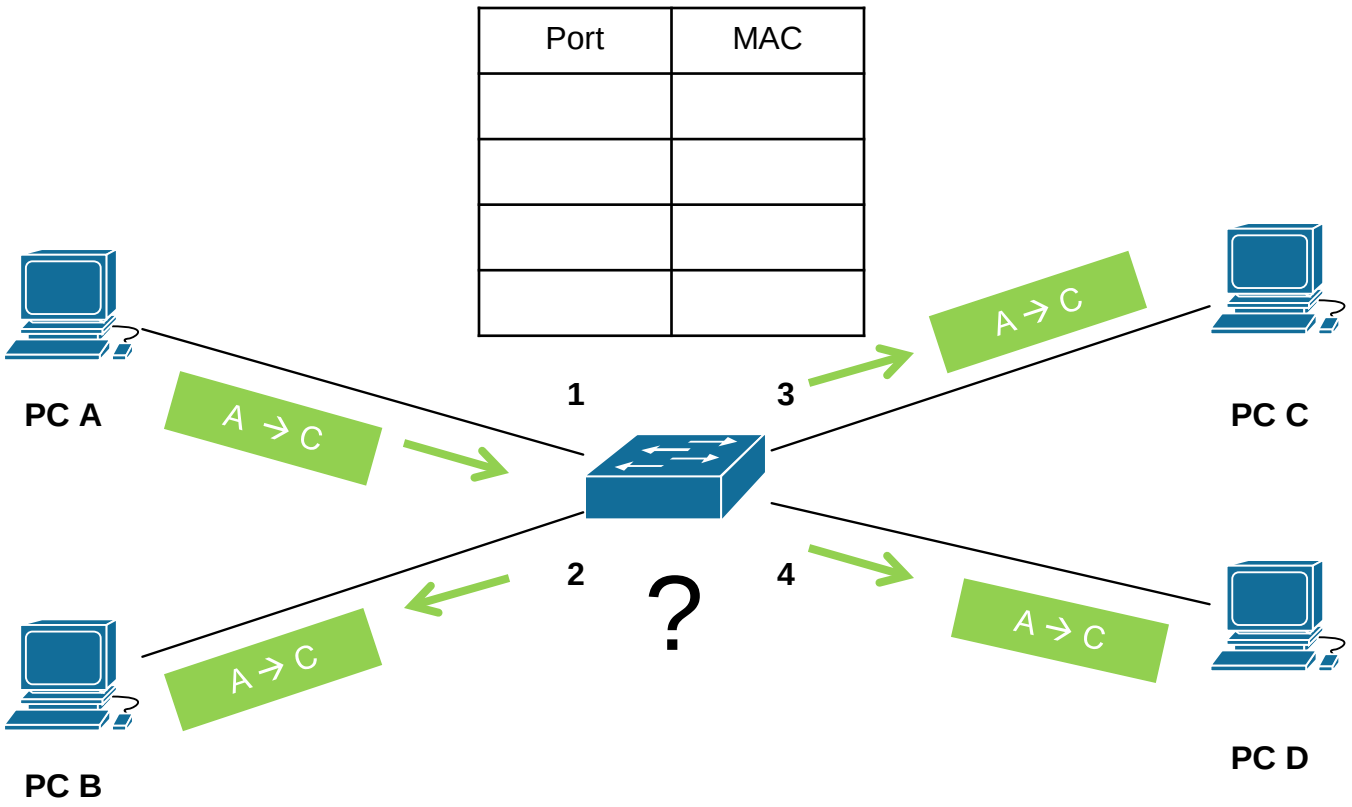
- TCP Small Servers (Echo, Chargen, Discard, Daytime)
- UDP Small Servers (Echo, Discard, Chargen)
- Finger
- Auto config
- Packet Assembler and Disassembler (PAD)
- BOOTP server
- Identification service
- NTP without authentication
- Source routing
- IP Proxy-ARP
- ICMP unreachables
- ICMP redirects
- Directed broadcast forwarding
- Maintenance Operation Protocol (MOP)

CAM Overflow Attack

Normal Operation ①

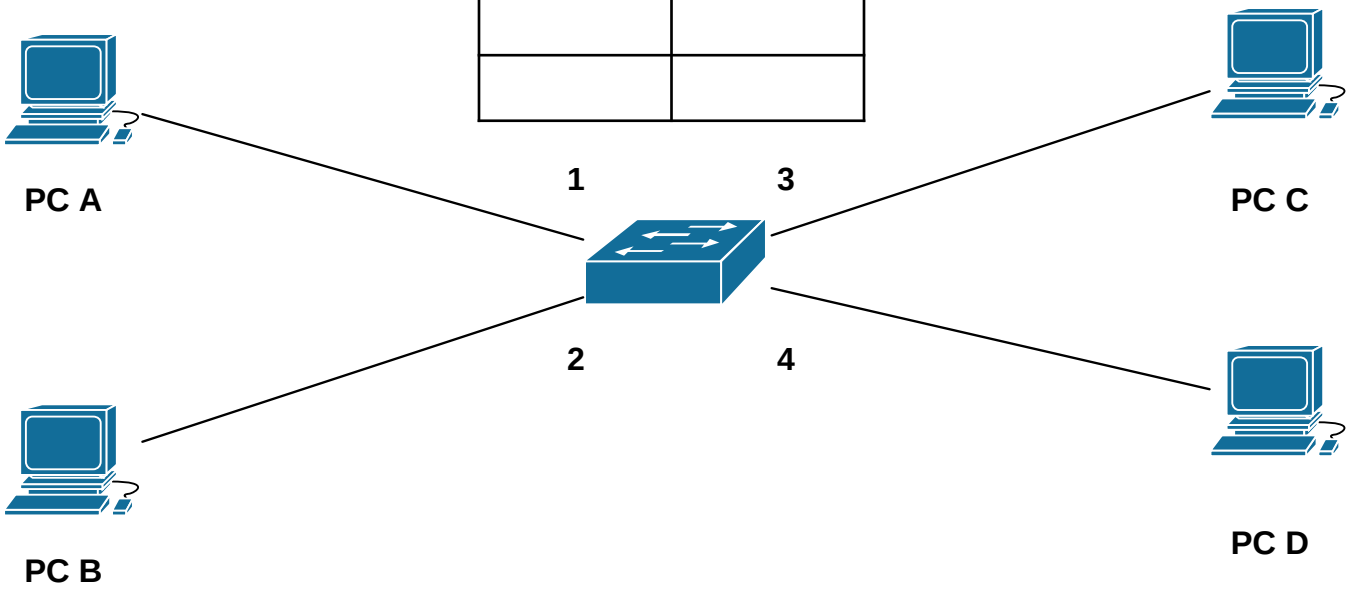


Normal Operation ②

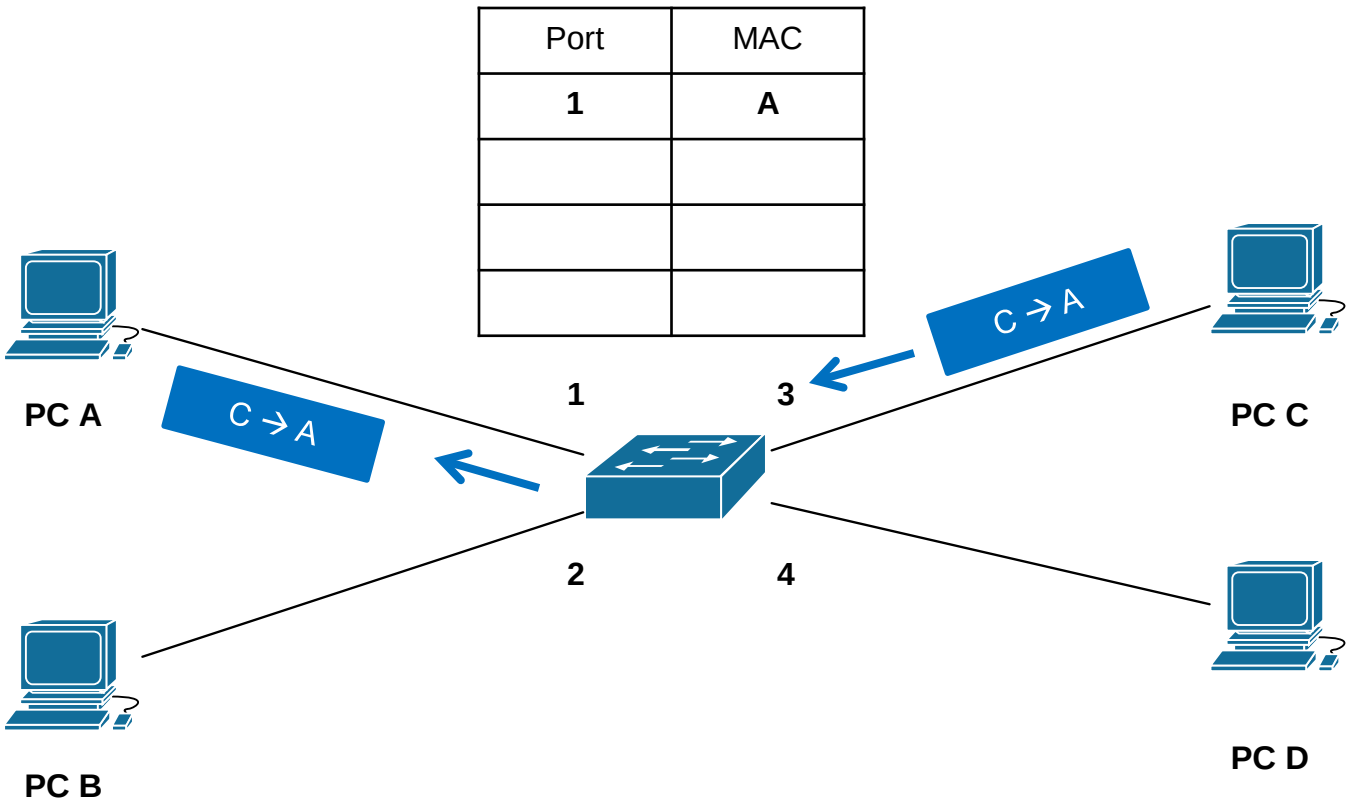


Normal Operation ③

Port	MAC
1	A

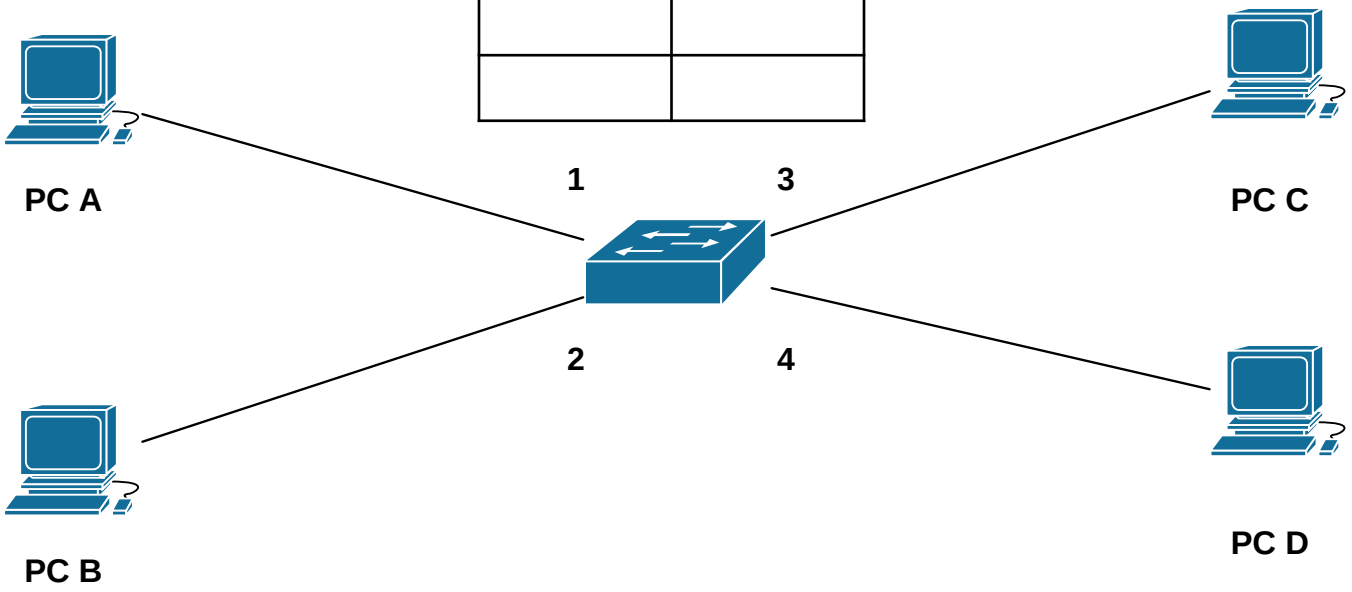


Normal Operation ④



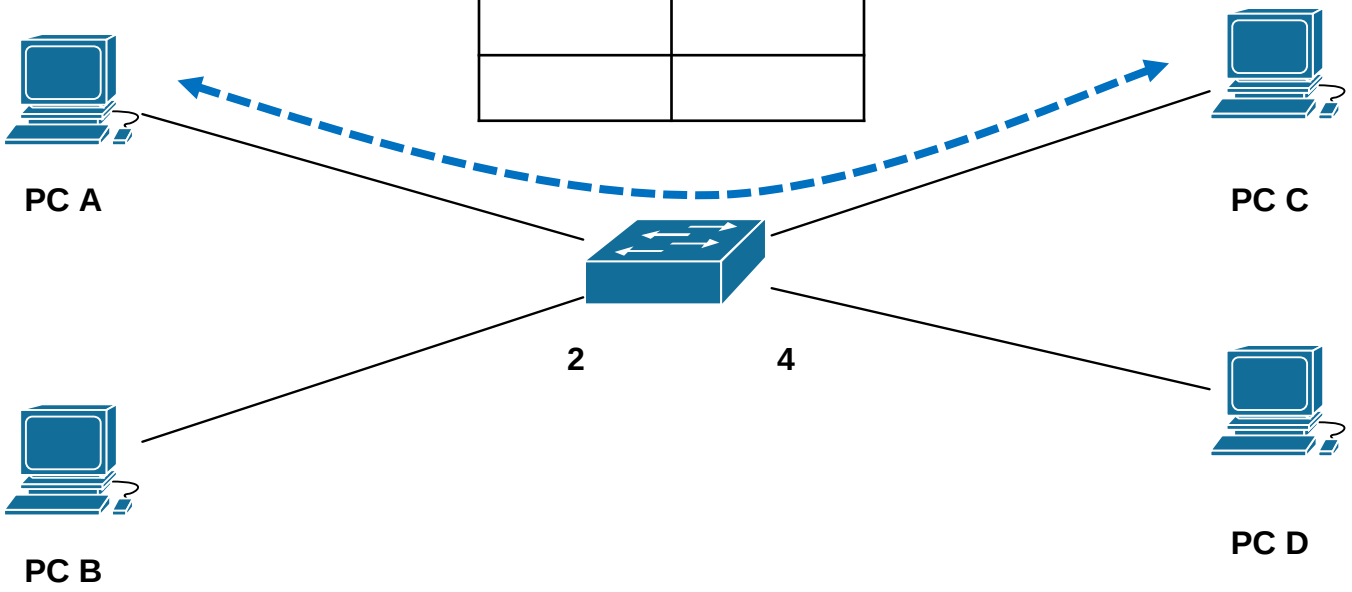
Normal Operation ⑤

Port	MAC
1	A
3	C

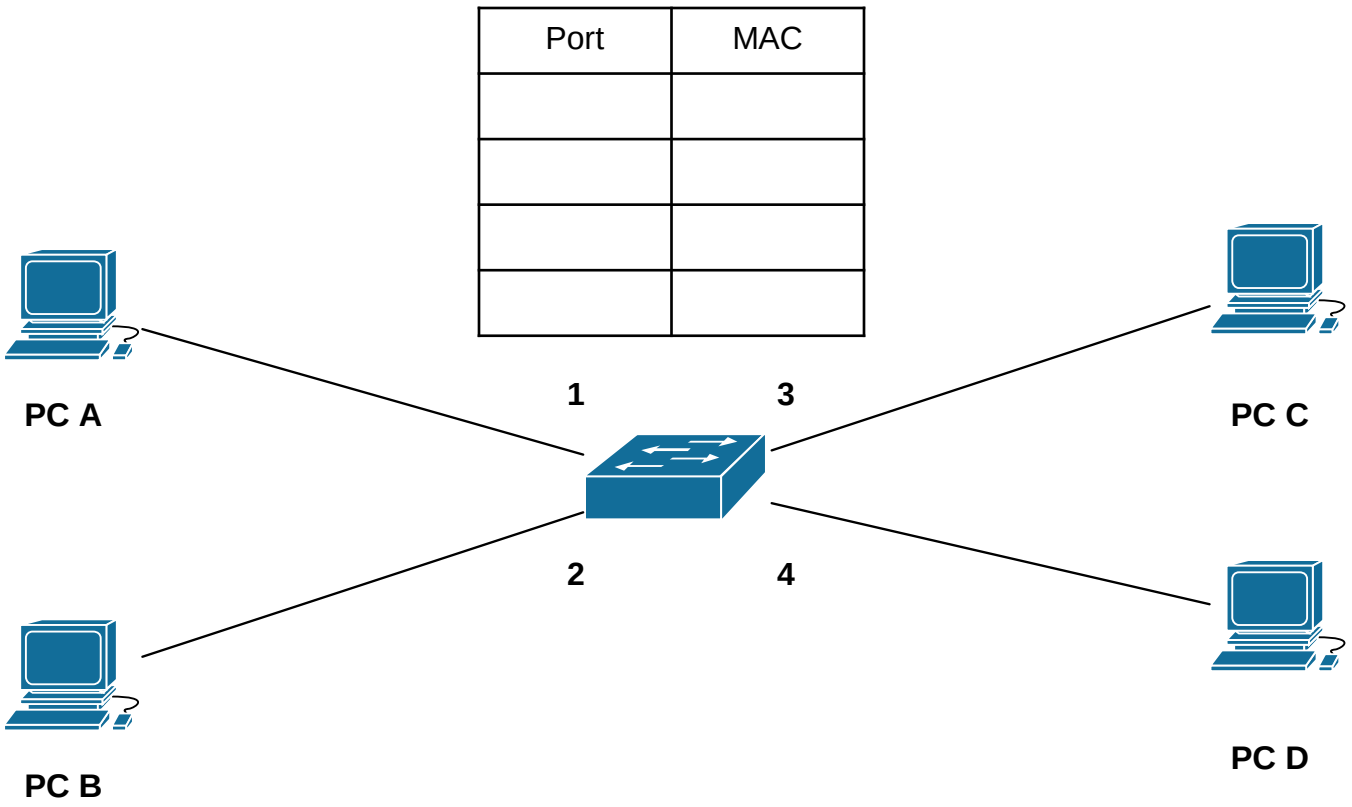


Normal Operation ⑥

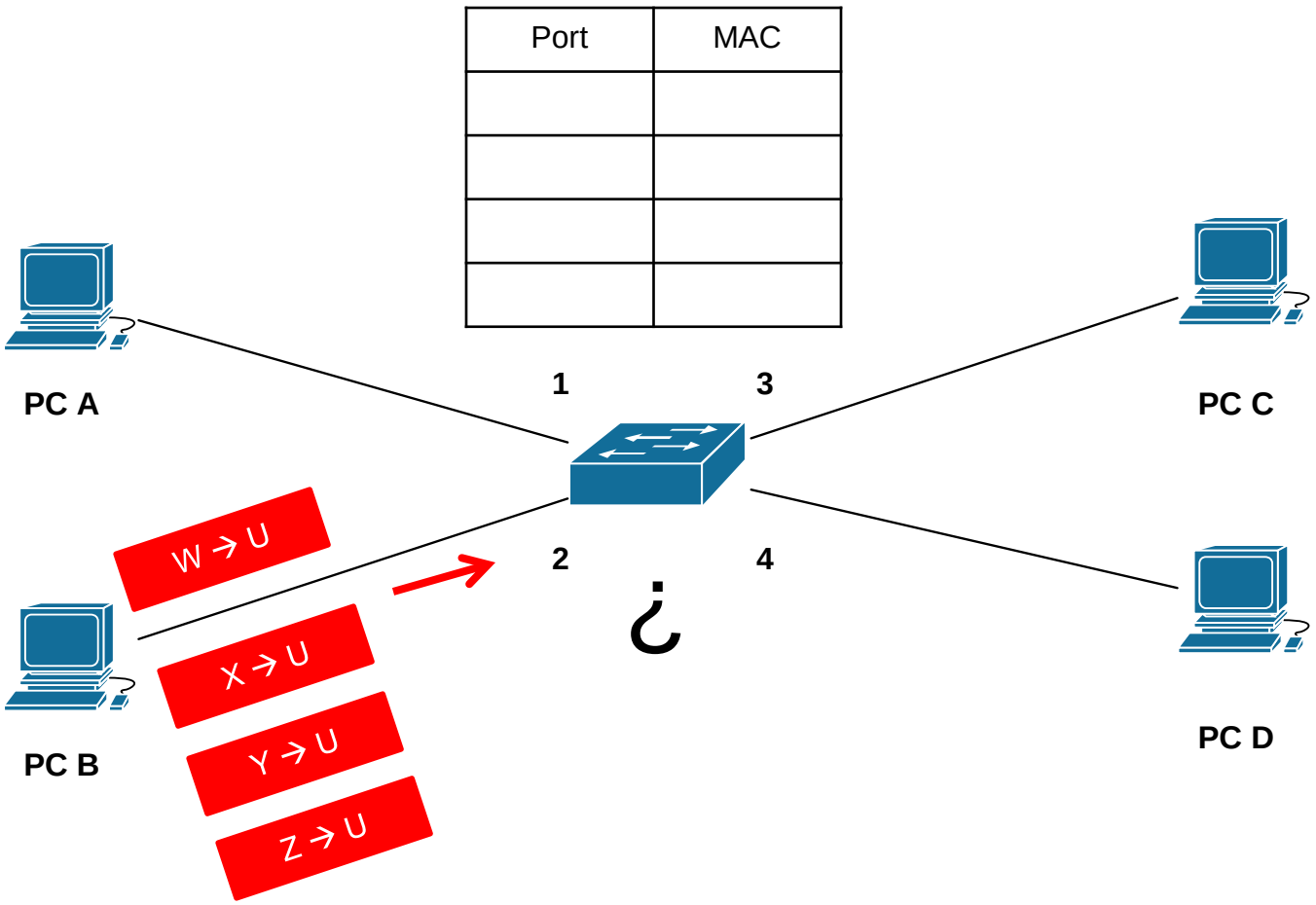
Port	MAC
1	A
3	C



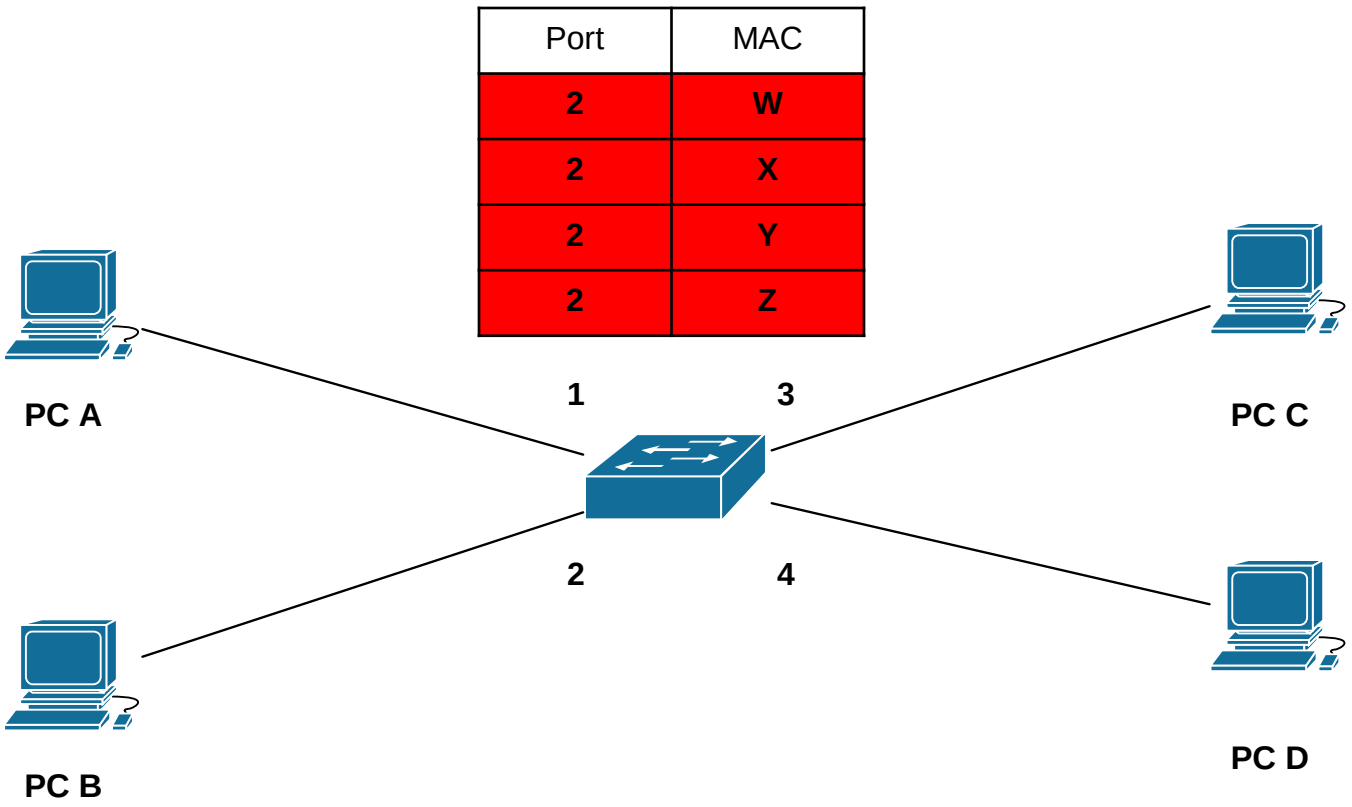
CAM Overflow Attack



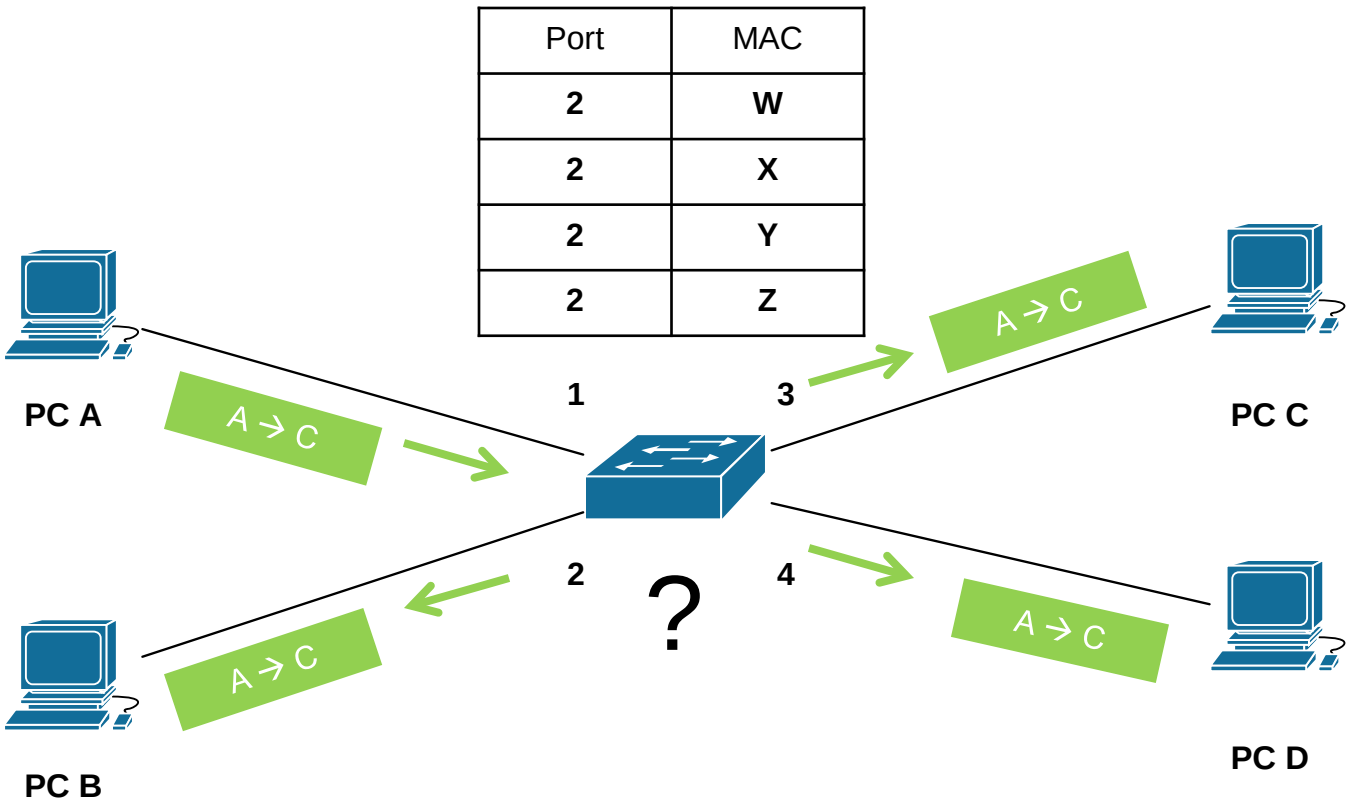
CAM Overflow Attack: Initial Flooding ①



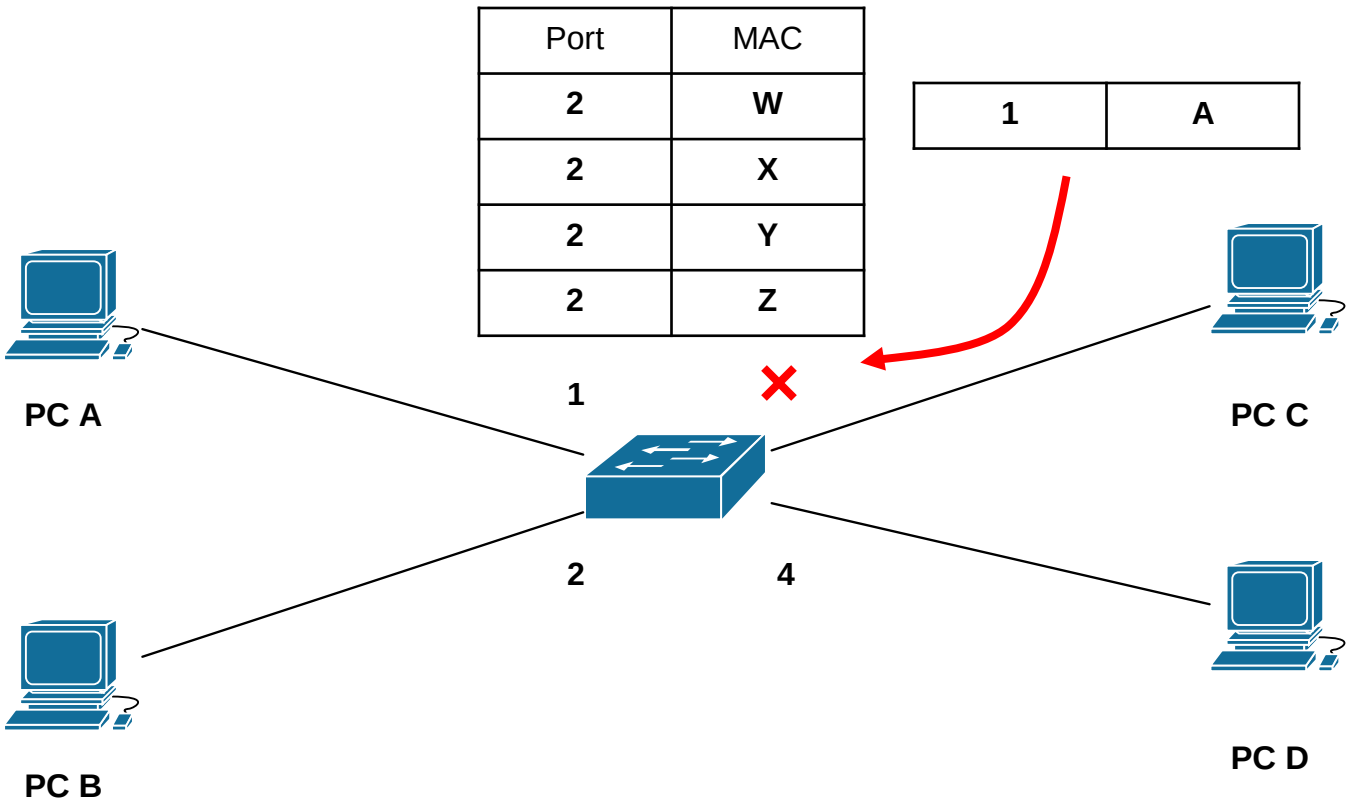
CAM Overflow Attack: Initial Flooding ②



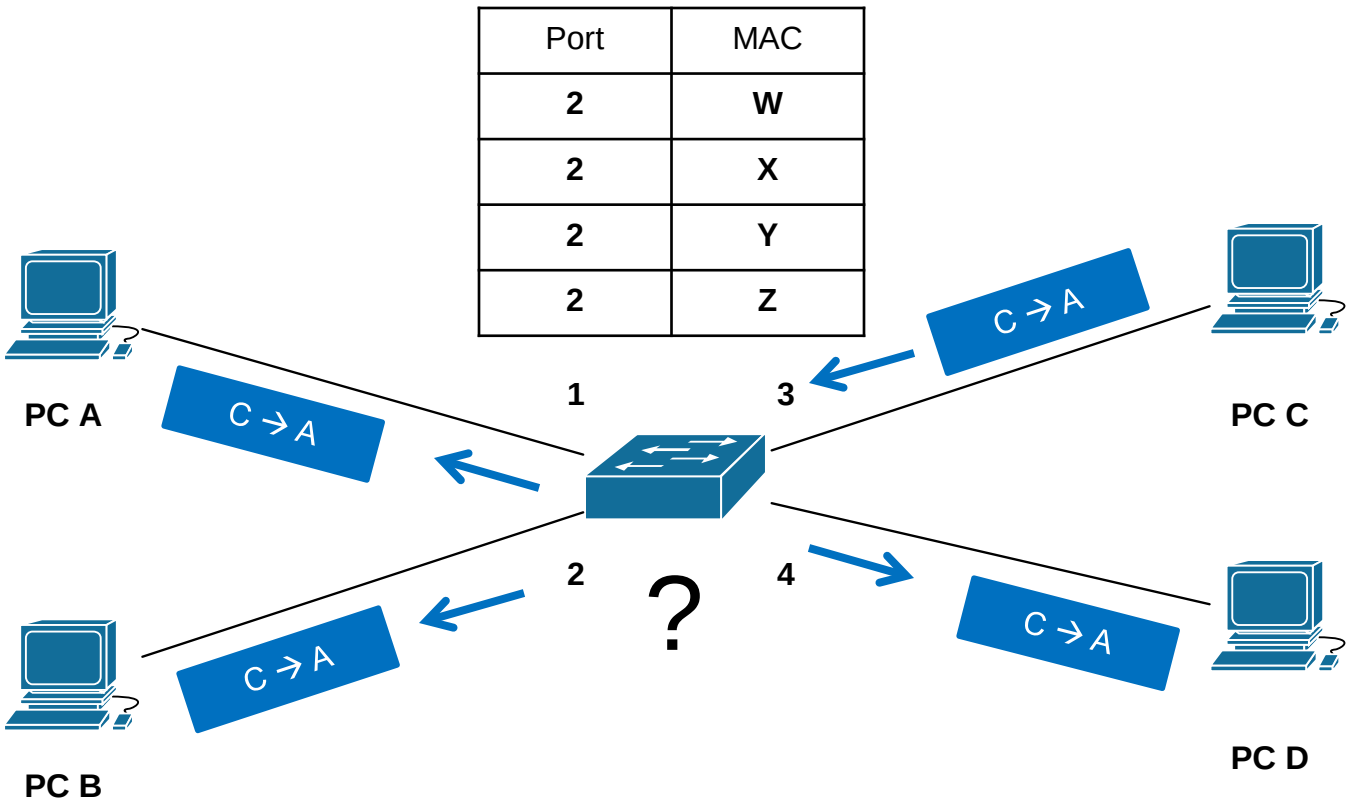
CAM Overflow Attack: Listening ①



CAM Overflow Attack: Listening ②



CAM Overflow Attack: Listening ③



CAM Table

- Apart from MAC, it stores VLAN identifier for target port and timout (Cisco default timeout is 300 seconds)
- CAM records CANNOT be pushed out!!!

Platform	CAM Size
Cisco Catalyst 2950	8 000
Cisco Catalyst 3560	12 000
Cisco Catalyst 3750	12 000
Linksys SRW224	4 000
Module to Cisco Catalyst 6500	128 000
HP ProCurve 2610	8 000
HP ProCurve 1400	8 000

Attack Properties

- Attack works only for limited amount of time (until CAM record timeout)
- When attacking, it is useful to generate twice more pseudo random MACs than it is CAM size
- Attack allows only **sniffing**
- *Tools*
  - Macof

```
root@ubuntu:~# macof –i eth0 –n 20000
c0:12:94:6d:2:34 17:84:37:5d:f7:f 0.0.0.0.17495 > 0.0.0.0.12933: S
321312800:321312800(0) win 512
```

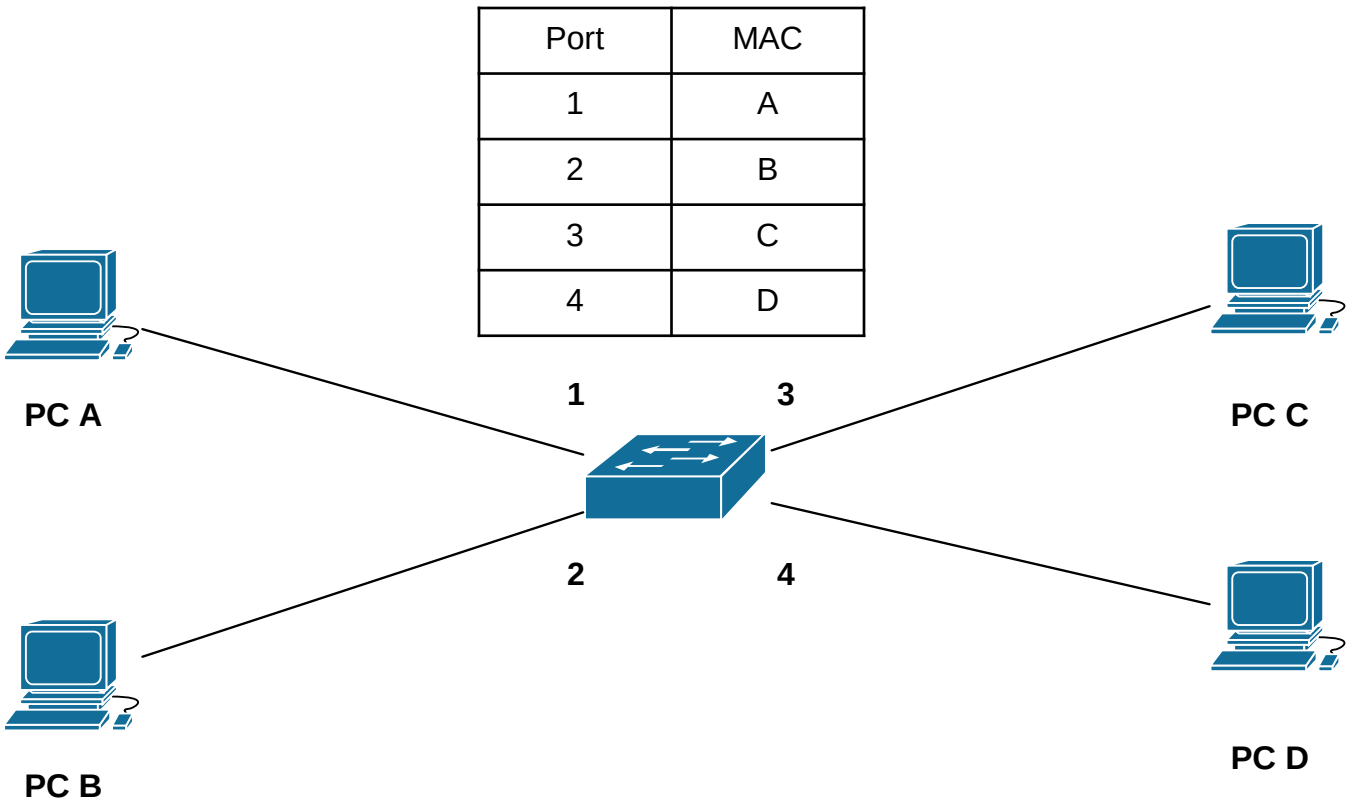
  - Scapy

Port Stealing

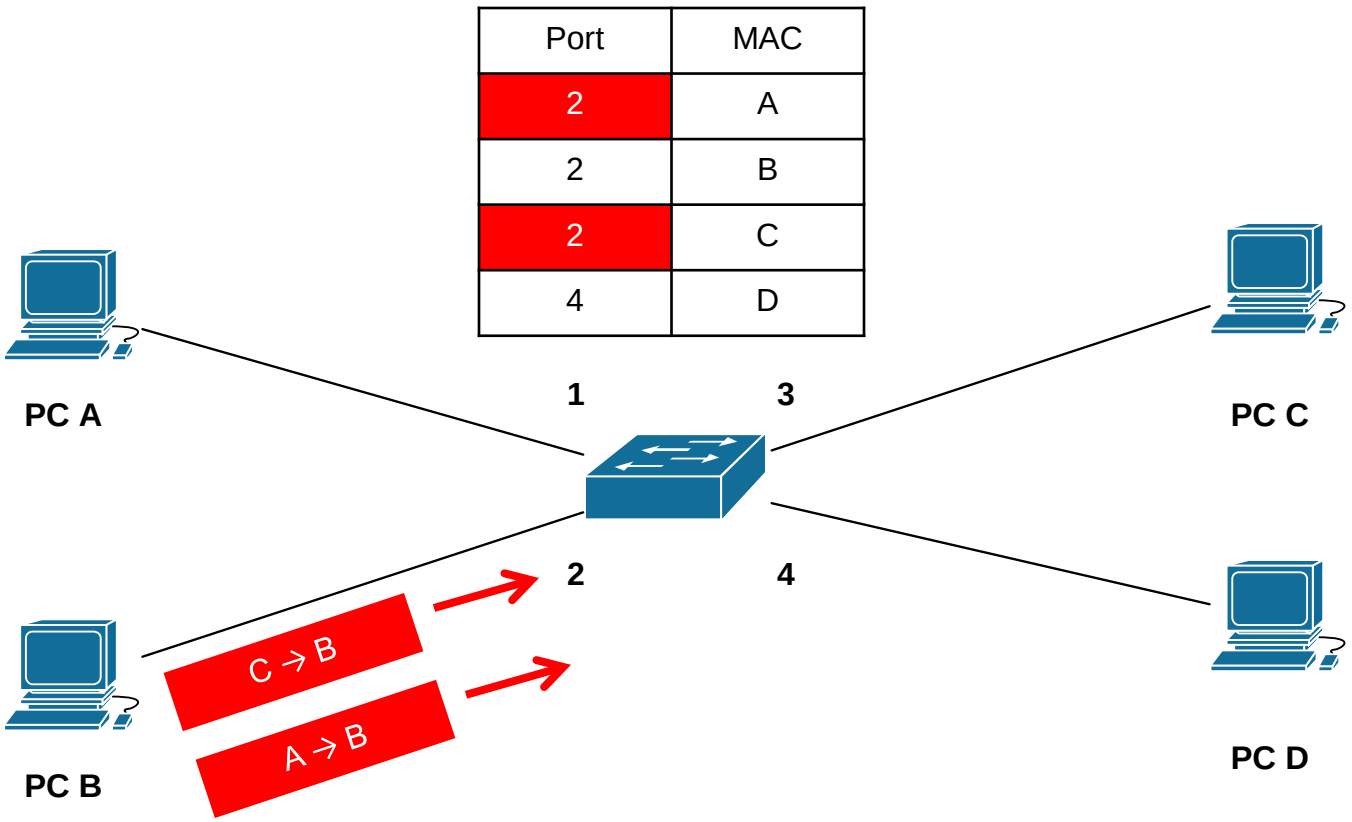
Port Stealing

- One of the few truly genuine Man-in-the-Middle attacks
- Non-reliable because of the race condition
- The goal is to rewrite binding of MAC address to certain port
  - MAC address cannot be pushed from CAM,...
  - ...but record is updated each time the MAC is learned
  - CAM table fooling is divided into three phases – listening (detection of victim's MAC), correction (optional rewrite using ARP), forwarding
  - The attack exists in two variants, according to their behavior let us call them **normal** and **aggressive port stealing**
- *Tools*
  - Ettercap implements normal port stealing

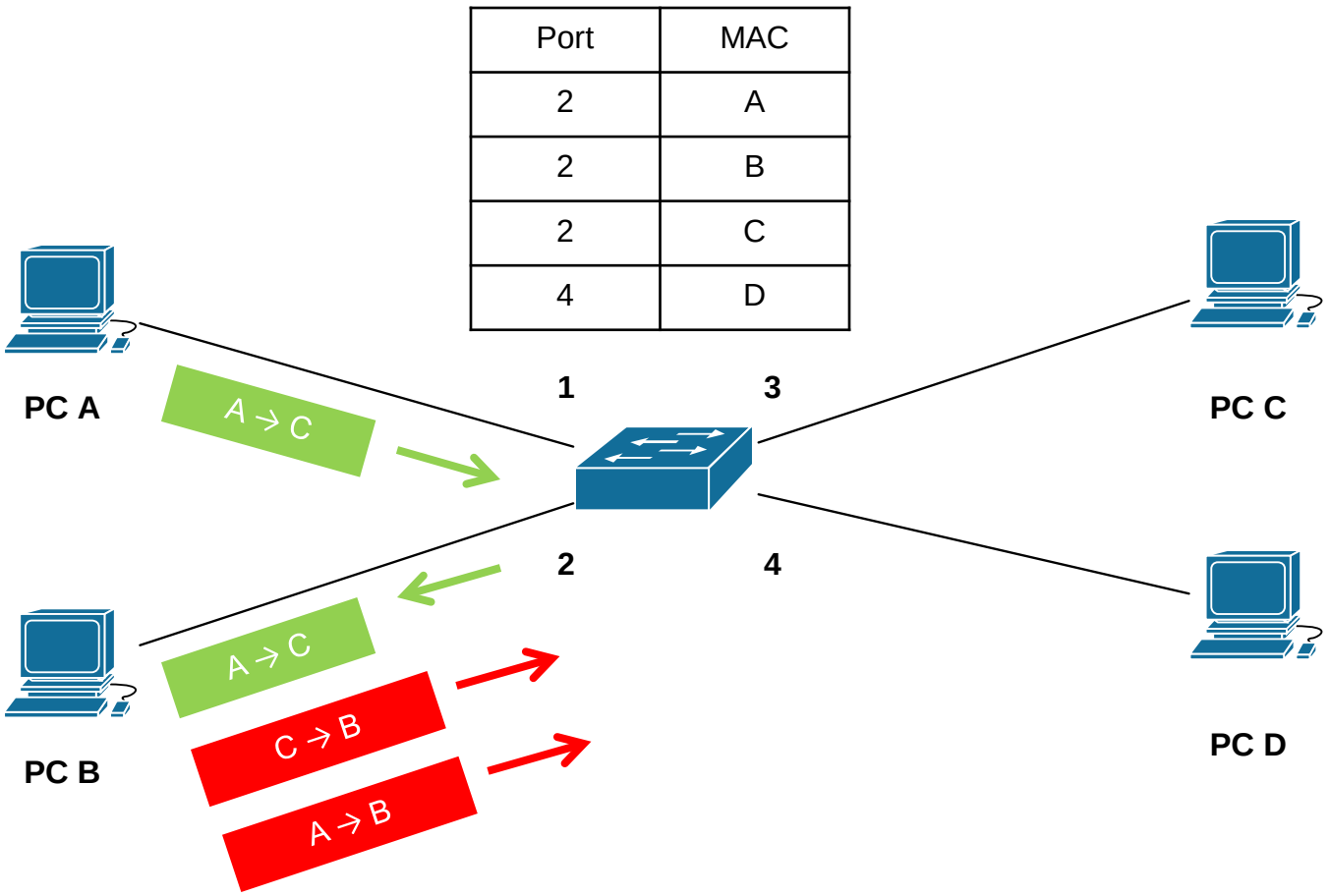
Normal Port Stealing Attack



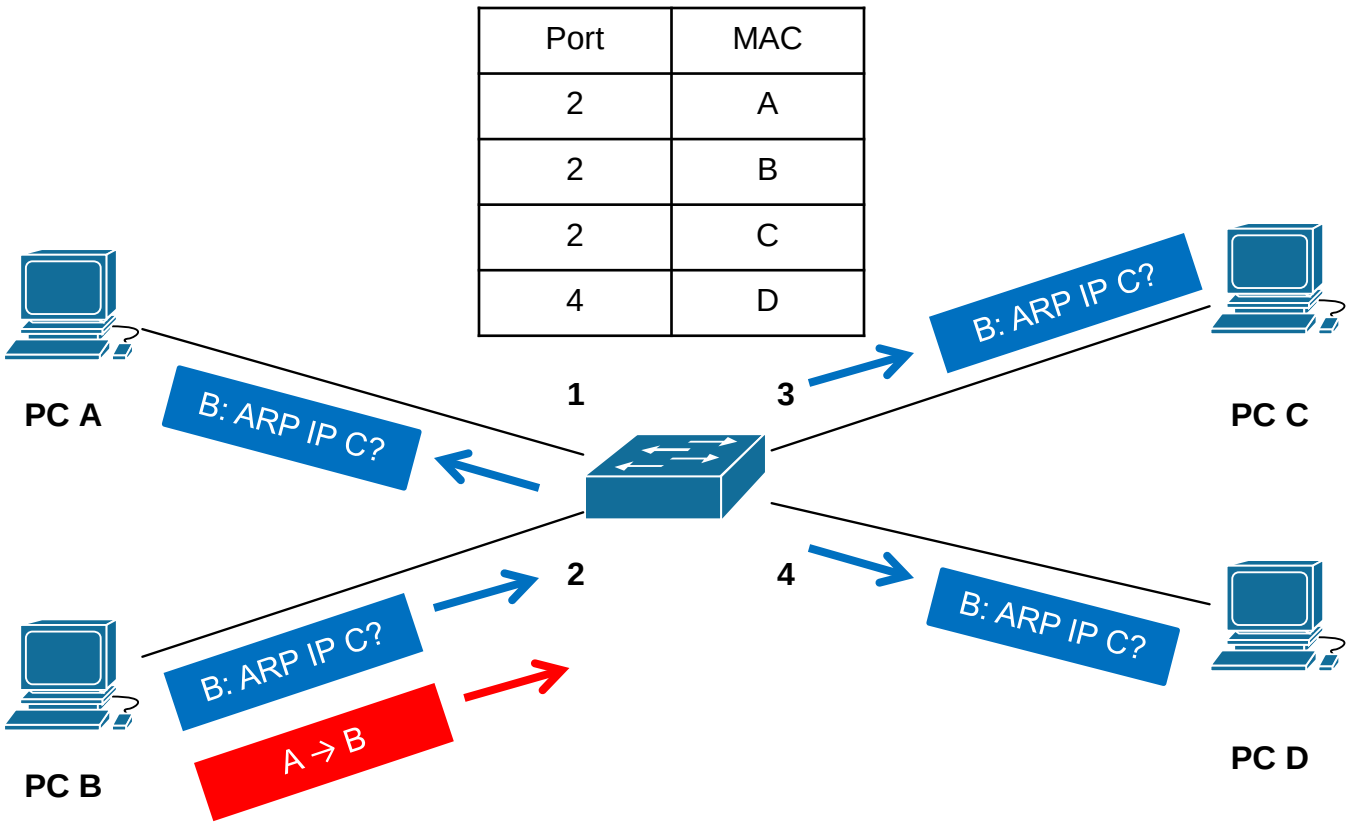
Normal Port Stealing Attack: Listening ①



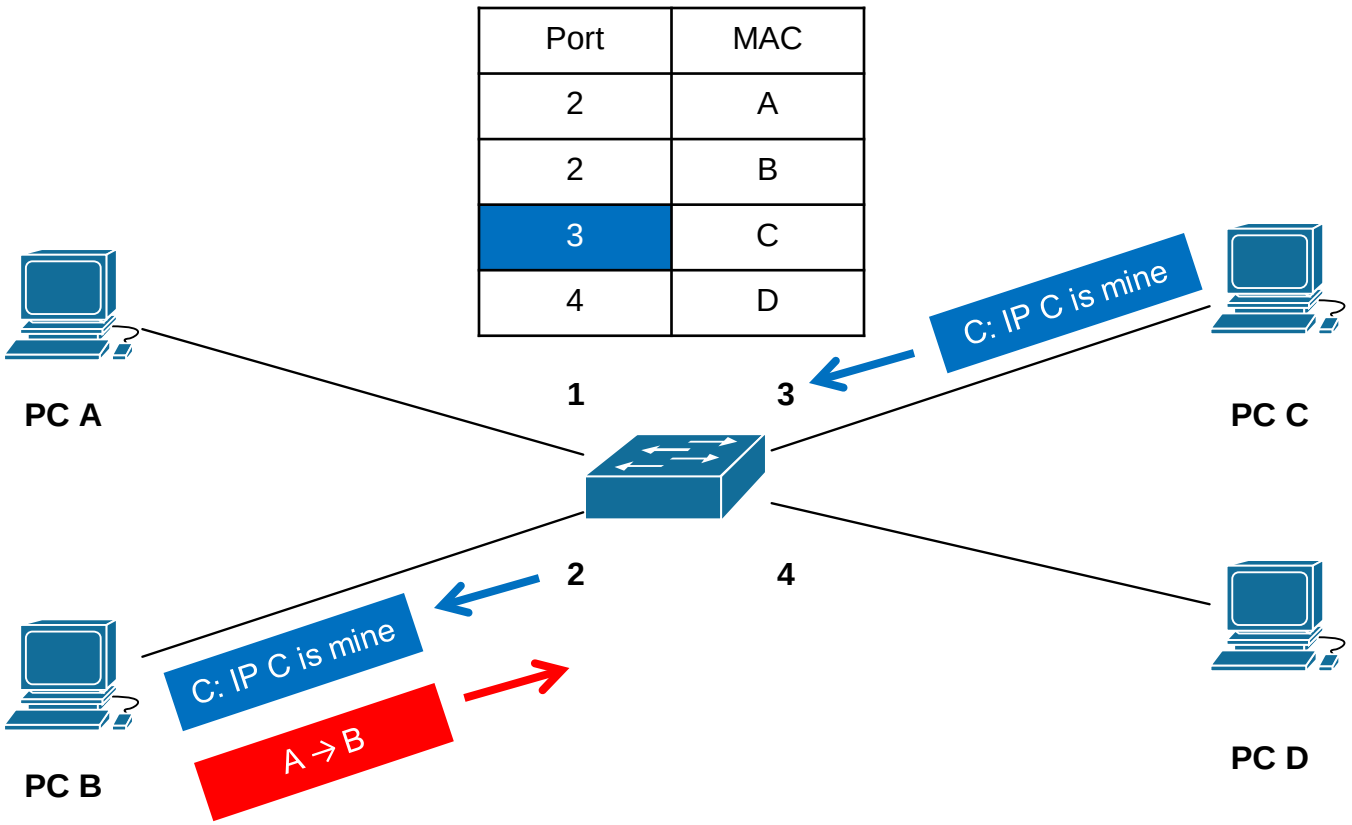
Normal Port Stealing Attack: Listening ②



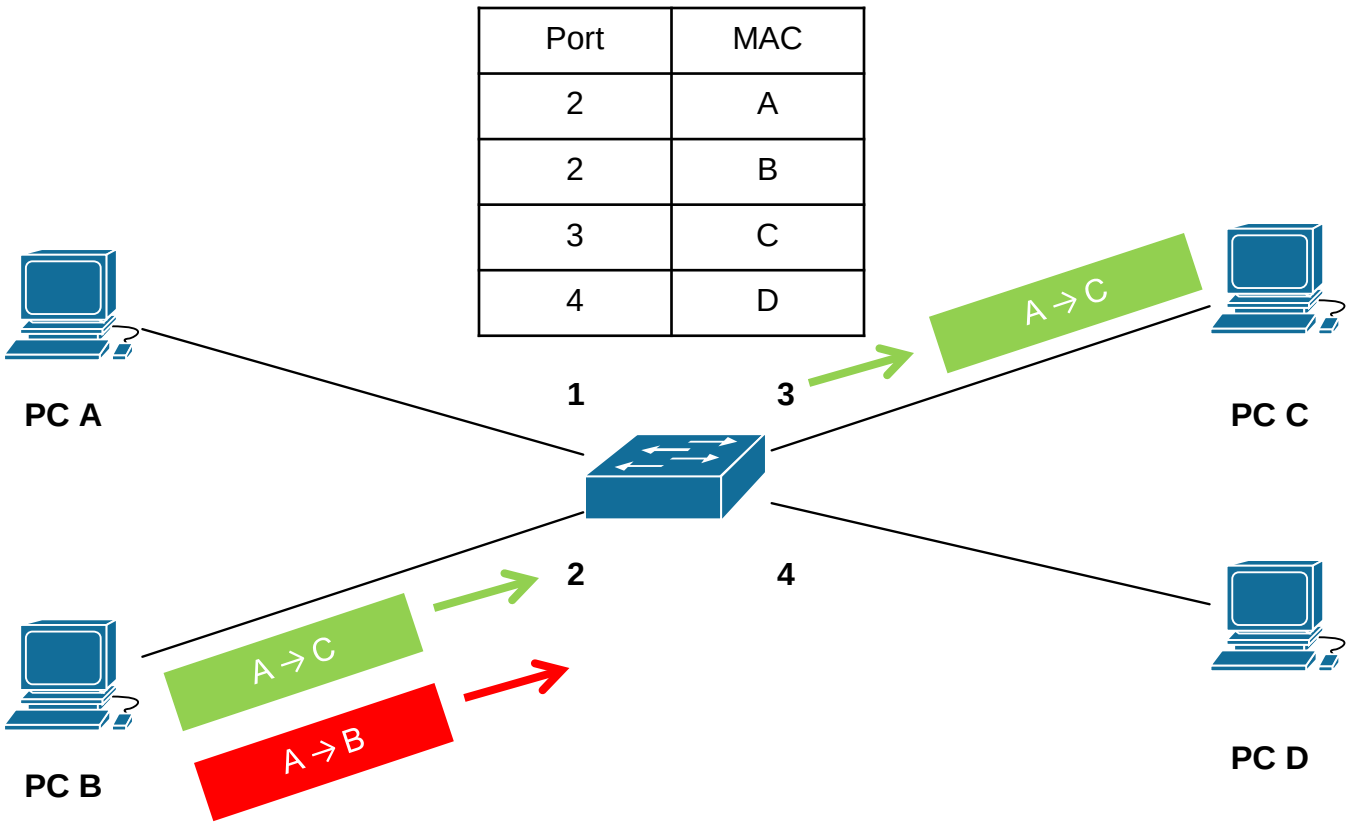
Normal Port Stealing Attack: Correction ①



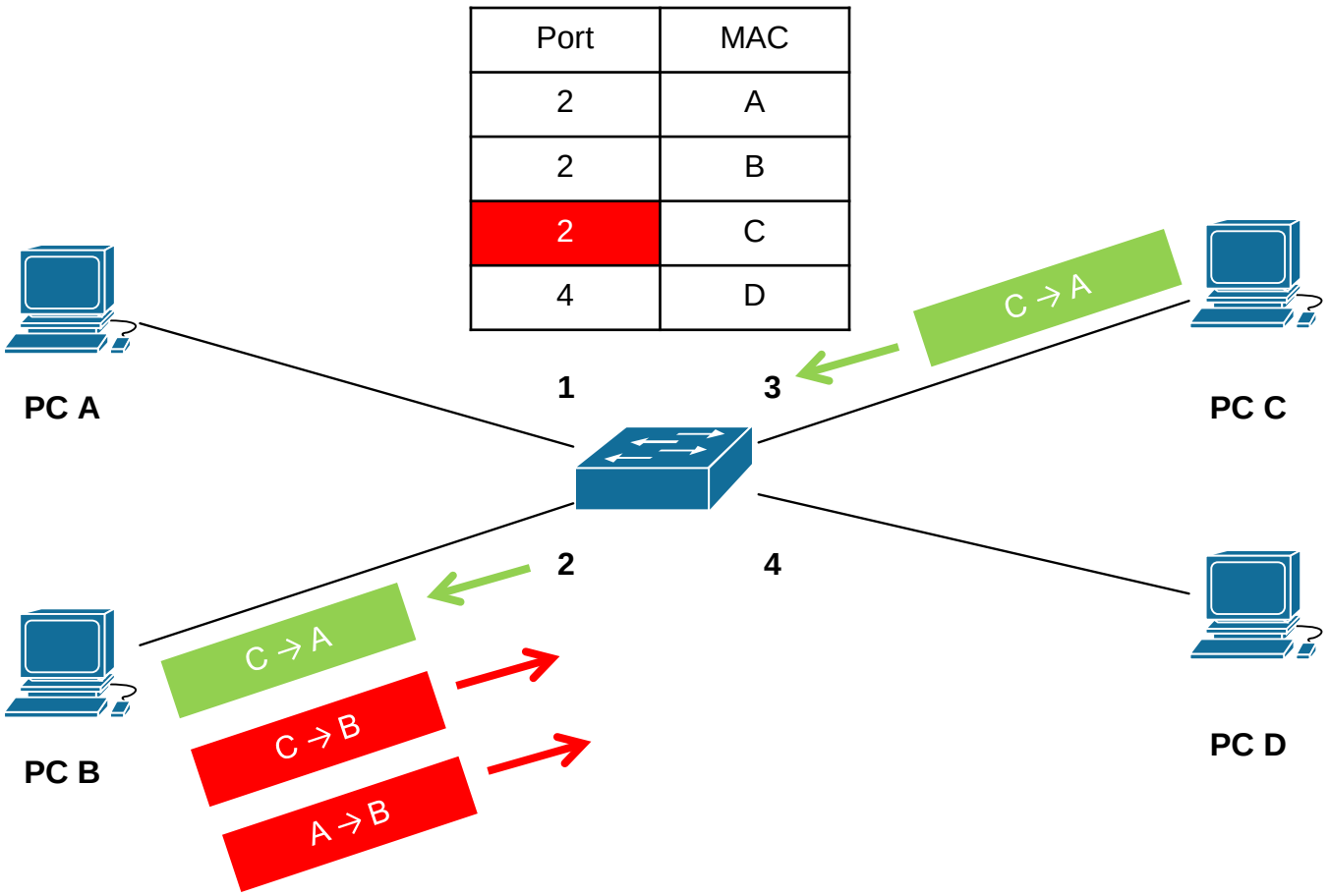
Normal Port Stealing Attack: Correction ②



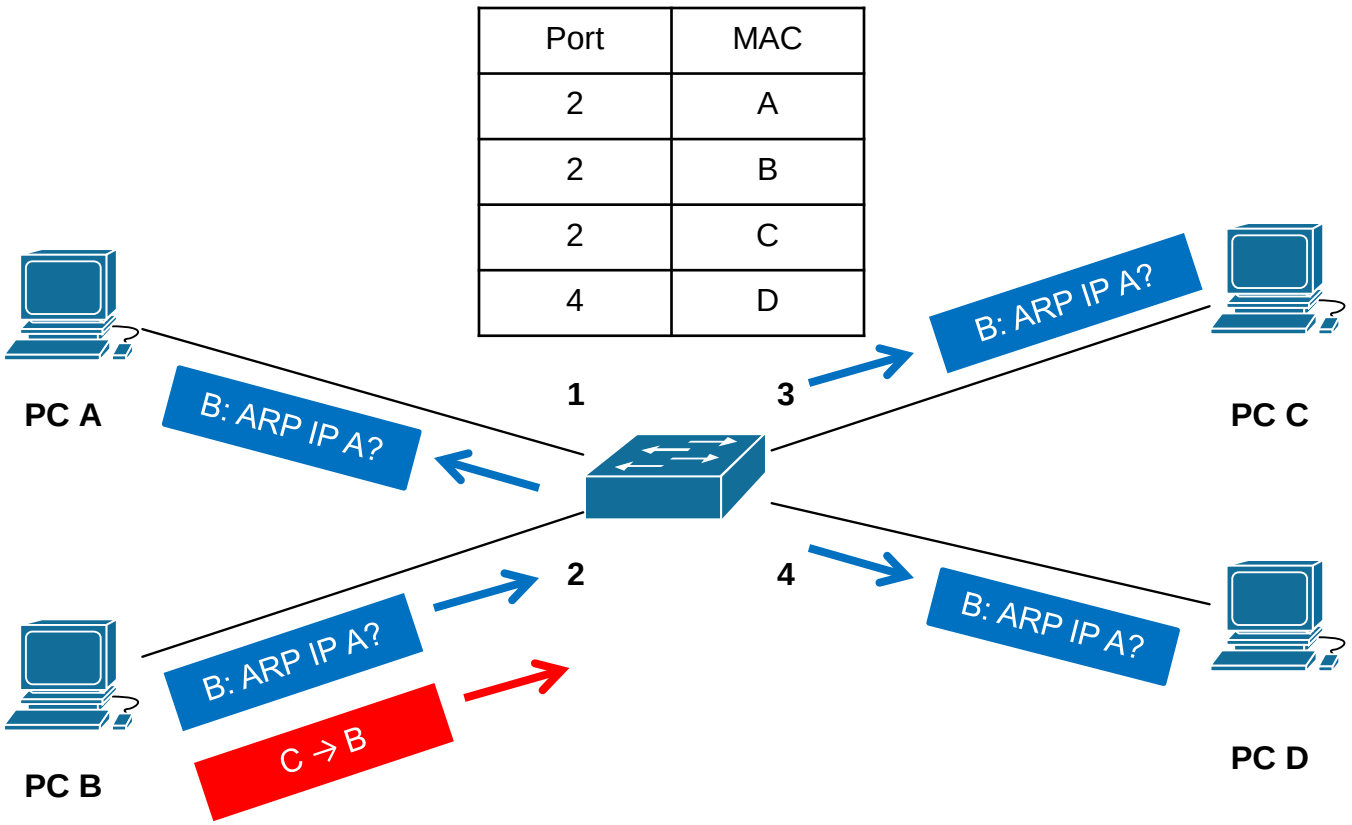
Normal Port Stealing Attack: Forwarding



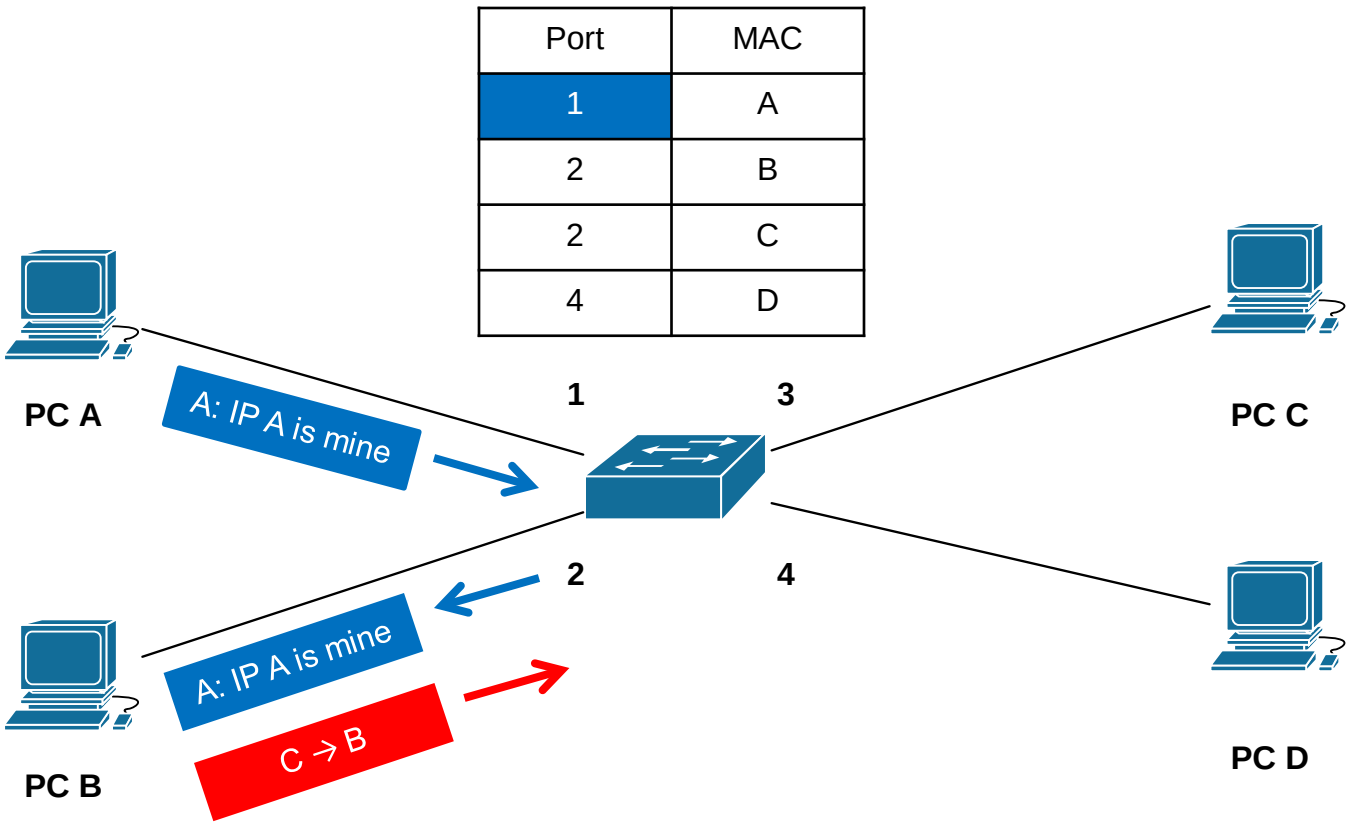
Normal Port Stealing Attack: Listening ①



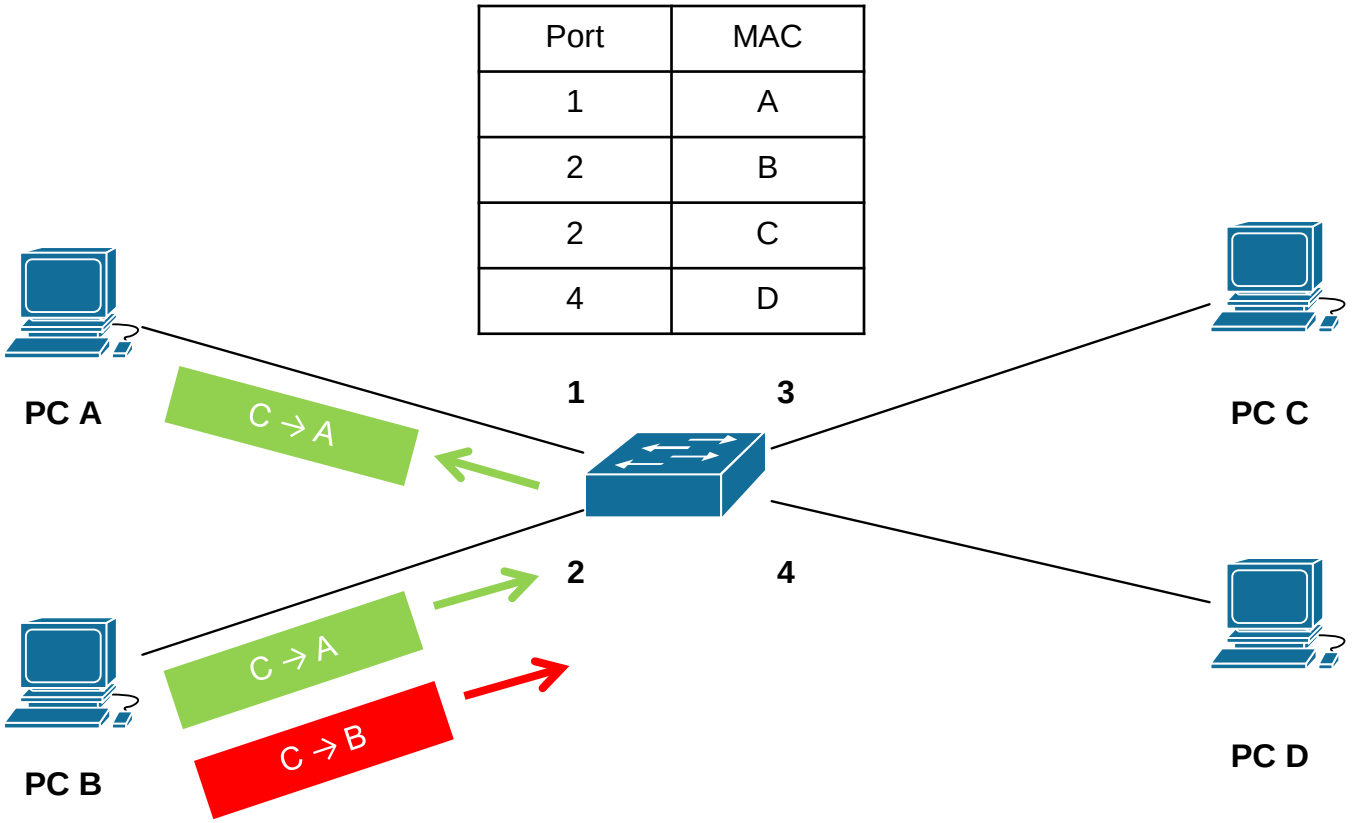
Normal Port Stealing Attack: Correction ①



Normal Port Stealing Attack: Correction ②



Normal Port Stealing Attack: Forwarding

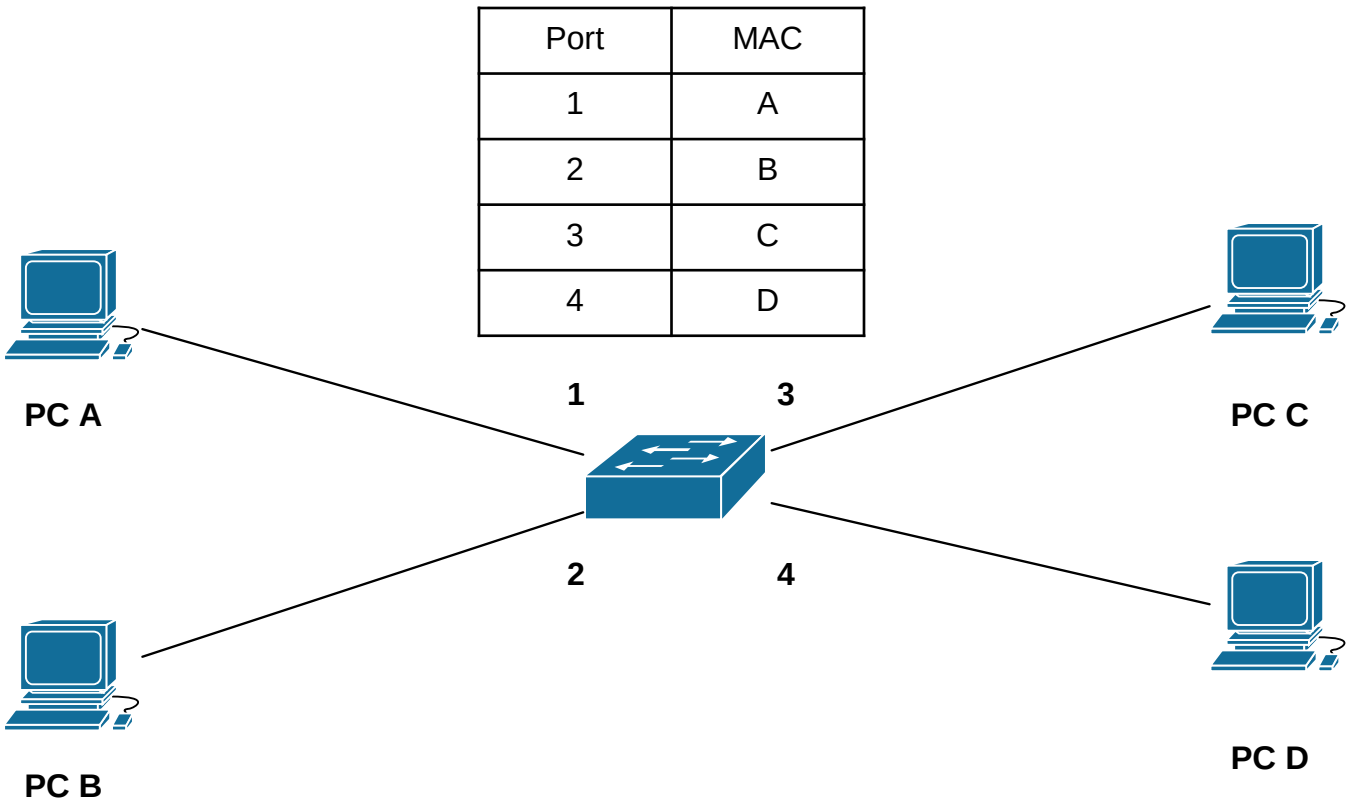


Limitations of Normal Port Stealing

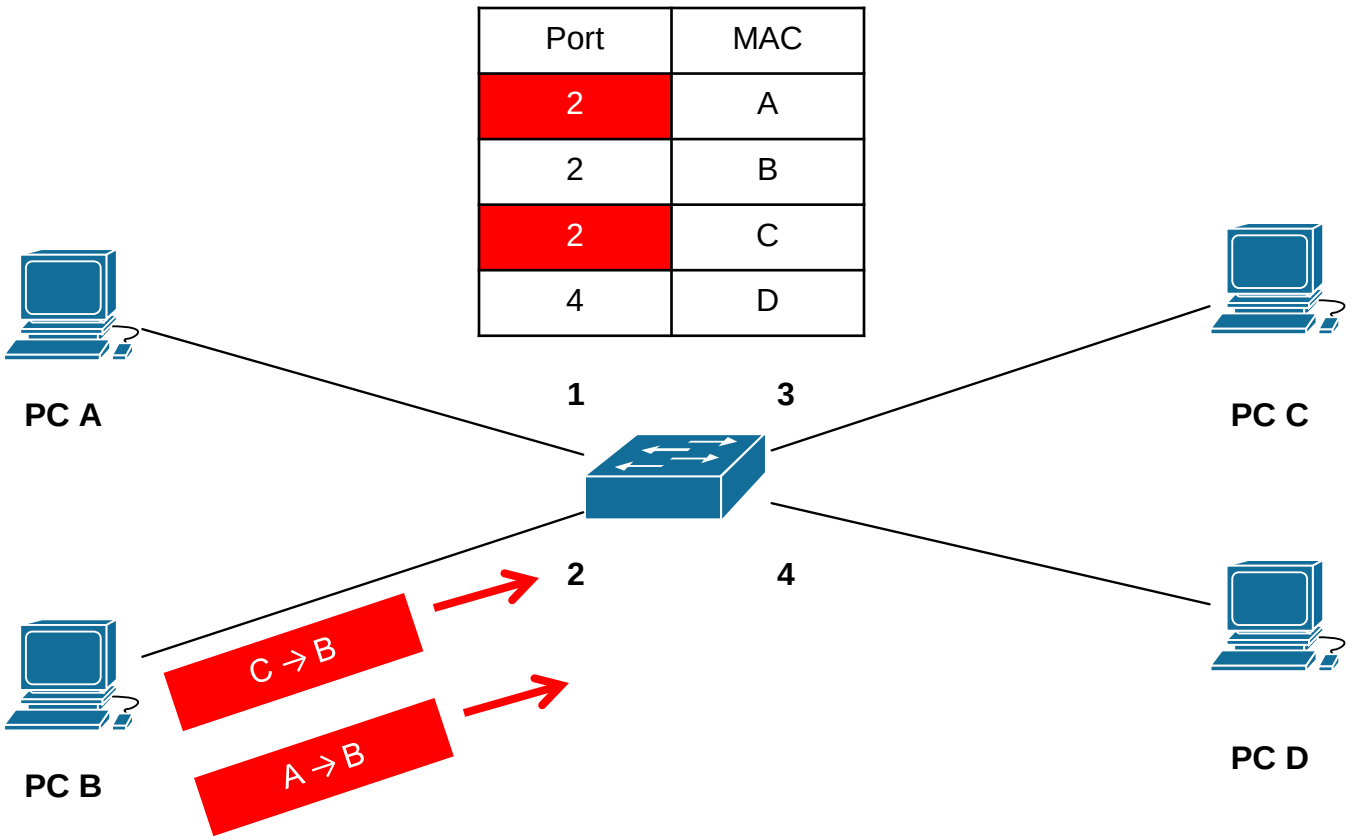
- *Maybe too complicated!*
- Suspectiable to message lost
- Slow because of: $t + ARP\ timeout$
- Buffer on attacker's side for correction phase
- Unreliable, because data may bypass attacker during correction and forwarding phase

- Hence, aggression mode removes unnecessary correction phase by using broadcast

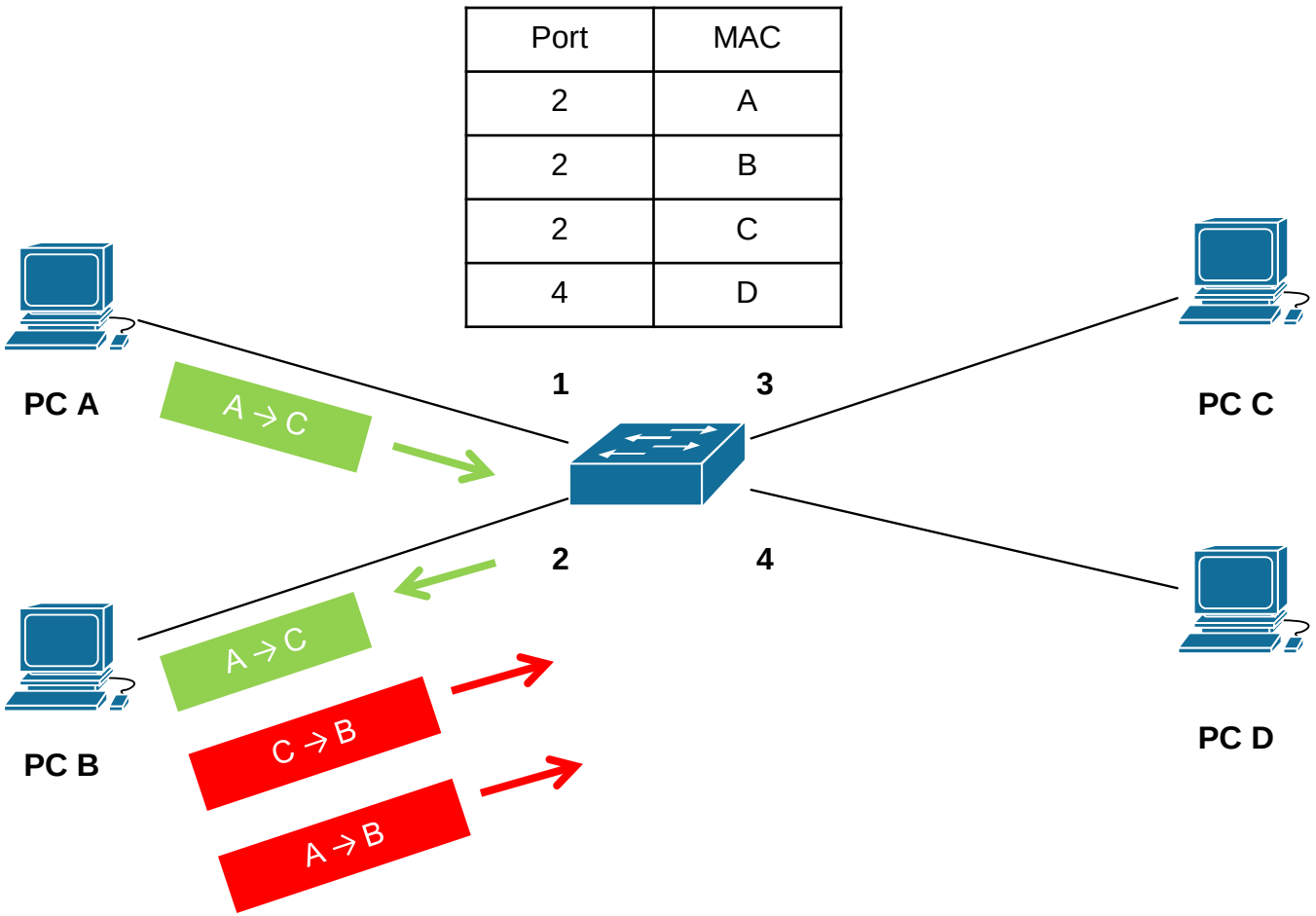
Aggressive Port Stealing



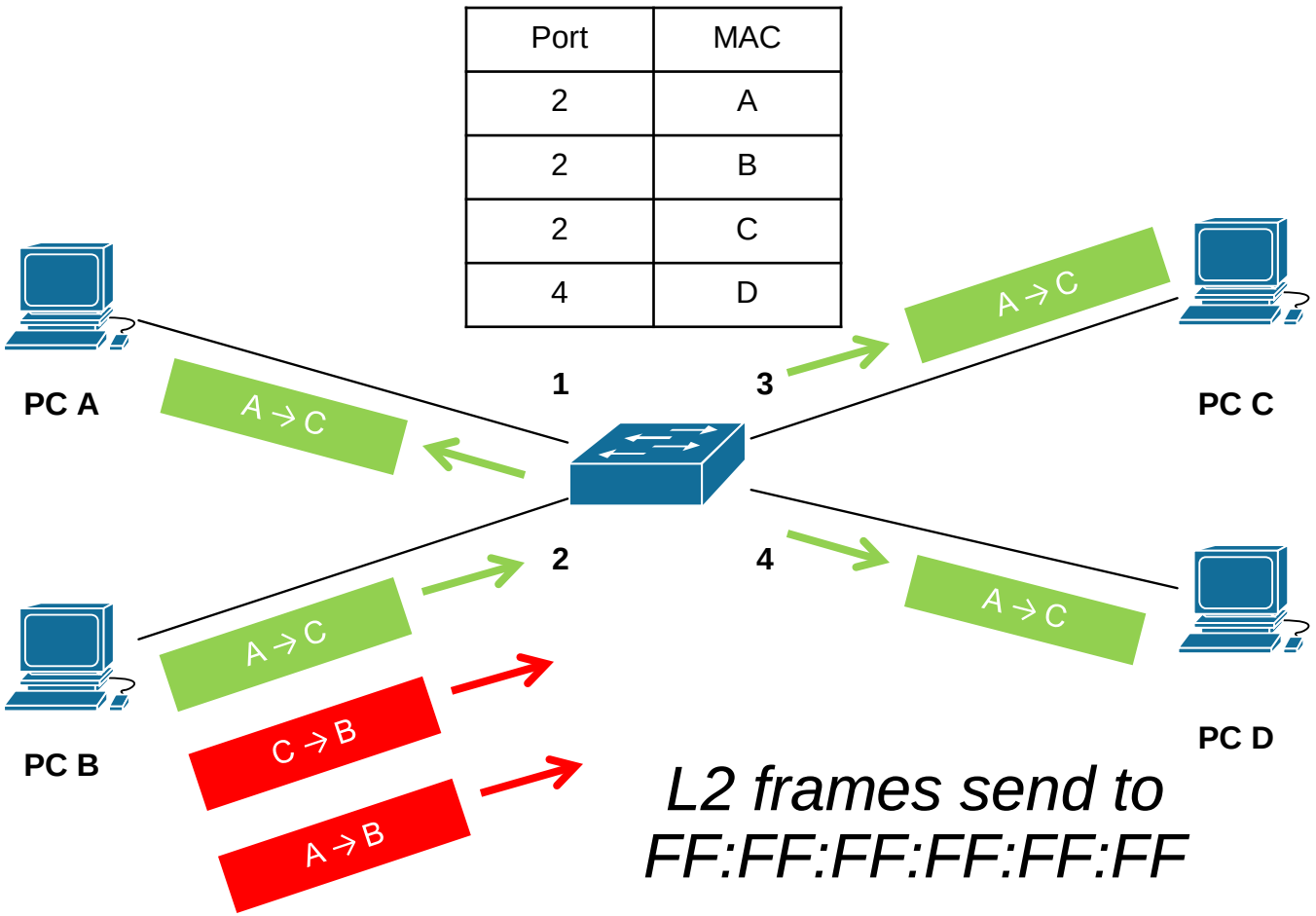
Aggressive Port Stealing: Listening ①



Aggressive Port Stealing: Listening ②



Aggressive Port Stealing: Forwarding



Port Security

Port Security

- Function allowing to define the maximum number of addresses on interface
  - Statically defined addresses are considered to be safe
  - Switch automatically adds any new unknown address to the CAM as dynamic or sticky record
  - IF a new record exceeds the MAC address limit THEN security violation is invoked

```
3w3d: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused
by MAC address dea3.2764.5d22 on port FastEthernet0/2.
```

- Following three types of security violation are available
  - **Protect** = the frame with the wrong MAC is discarded
  - **Restrict** = the frame with the wrong MAC is discarded and incident is reported (console message, SMTP trap, Syslog)
  - **Shutdown** = the interface is shutdowned and turned into err-disabled state

Configuration

1) Enable port security:

```
Switch(config-if)# switchport port-security
```

2) Set a maximum number of MAC addresses that will be allowed on this port. The default is one:

```
Switch(config-if)# switchport port-security maximum value
```

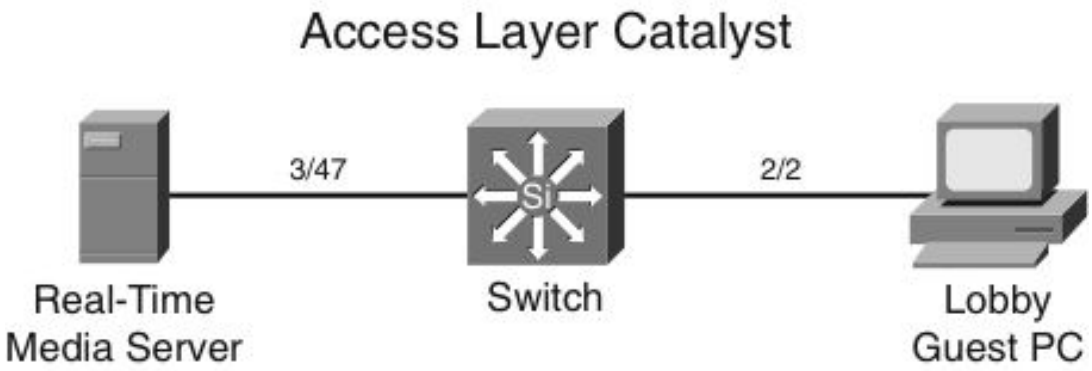
3) Specify which MAC addresses will be allowed on this port:

```
Switch(config-if)#
   switchport port-security mac-address mac-address
```

4) Define what action an interface will take if a non-allowed MAC address attempts access:

```
Switch(config-if)# switchport port-security violation
{shutdown | restrict | protect}
```

Example




```
Switch(config)# interface FastEthernet 3/47
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address 0000.0000.0008
Switch(config-if)# switchport port-security maximum 1
Switch(config-if)# switchport port-security aging time 2
Switch(config-if)# switchport port-security aging static
Switch(config-if)# switchport port-security violation restrict
Switch(config)# interface FastEthernet 2/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security maximum 1
Switch(config-if)# switchport port-security aging time 2
Switch(config-if)# switchport port-security aging static
Switch(config-if)# switchport port-security violation shutdown
```

Verification

```
switch# show port-security interface fastethernet0/1
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Restrict
Aging Time : 60 mins
Aging Type : Inactivity
SecureStatic Address Aging : Enabled
Maximum MAC Addresses : 2
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Last Source Address:Vlan : 001b.d513.2ad2:5
Security Violation Count : 0

switch# show port-security address
        Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address    Type            Ports   Remaining Age
                                                    (mins)
----    -----------    ----            -----   -------------
2       001b.d513.2ad2 SecureDynamic   Fa0/1   60 (I)
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 6144
```

Port Security Remarks

- `IF` Port Security is configured and enabled `THEN` all learned addresses in CAM are converted to allowed addresses
- Command **`no switchport port-sec mac-addr sticky`** removes all sticky MAC address from configuration
  - However, all of them remain in CAM as dynamic secure records until expiration
- Interfaces with Voice VLAN requires port security maximum to be at least 2

Blocking Unicast Flooding

- Cisco Catalyst switches can restrict flooding of unknown multicast MAC-addressed traffic on a per-port basis, in addition to restricting flooding of unknown unicast destination MAC addresses.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet 3/22
Switch(config-if)# switchport block unicast
Switch(config-if)# switchport block multicast
```

VTP Attack

VTP Attack

- VTP attack aims to disrupt integrity of VLAN database
- VTP is naive protocol until version 3
- Mitigation
  - Secure domain with non-trivial VTP password
  - Do not connect end-stations to trunk ports
  - `IF` it is necessary to connect server to a trunk port `THEN` secure this port with MAC ACL
    - Destination MAC 01-00-0C-CC-CC-CC
    - Type 0x2003

Double-tagging Attack

Double-tagging

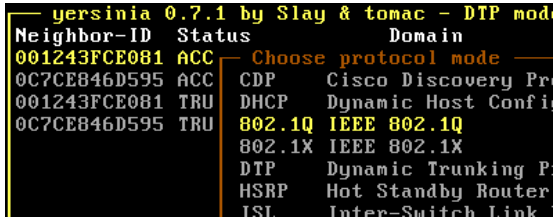
Normal Frame

Source MAC (6 Bytes)	Destination MAC (6 Bytes)	Type/Length (2 Bytes)	Data	FCS (4 Bytes)

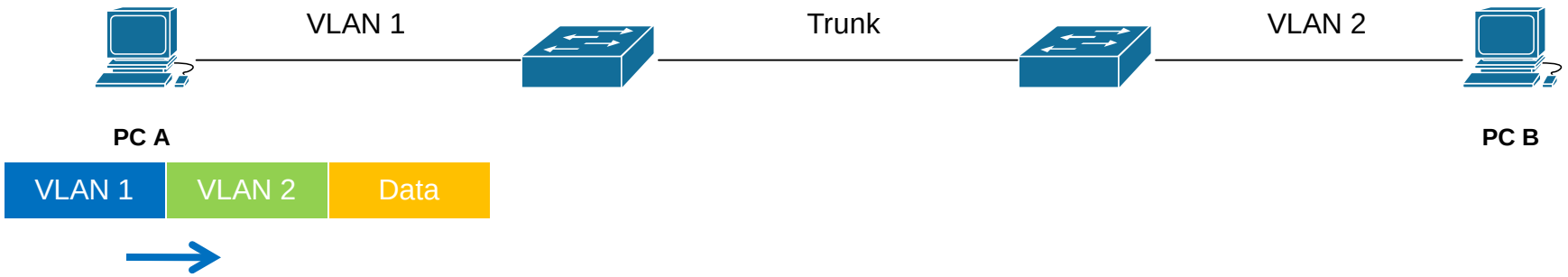
Tagged Frame

Source MAC (6 Bytes)	Destination MAC (6 Bytes)	Type/Length (2 Bytes)	802.1Q Tag (4 Bytes)	Data	FCS (4 Bytes)

- Available only in 802.1Q networks due to the configuration errors
- One-way only (from attacker towards victim)
- Requirements
  a. Attacker and victim are connected to different switches
  b. Trunk between switches uses 802.1q protocol
  c. Native VLAN of trunk is the same as access VLAN of attacker
  d. Tagging of native VLAN is disabled
- Tools
  - Yersinia
  - Scapy



Double-tagging: Principle ①



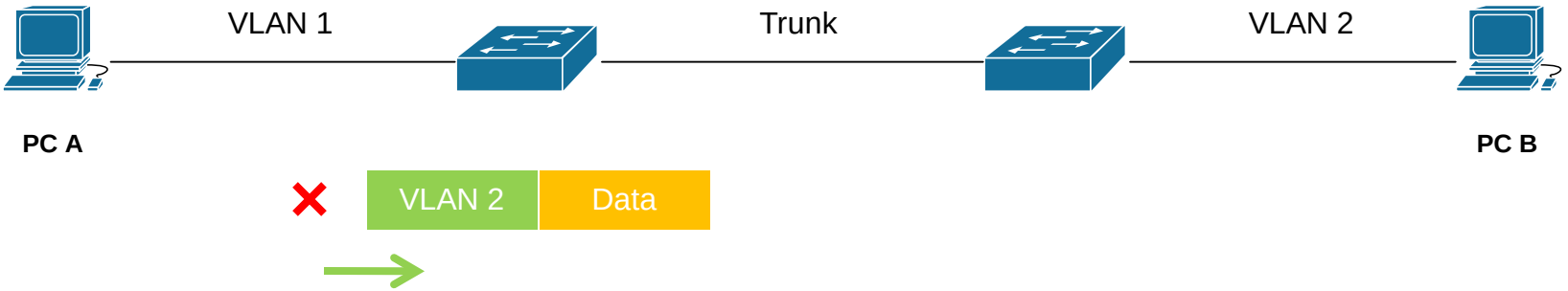
Src. MAC: A

Dst. MAC: B

Src. IP: A

Dst. IP: B

Double-tagging: Principle ②



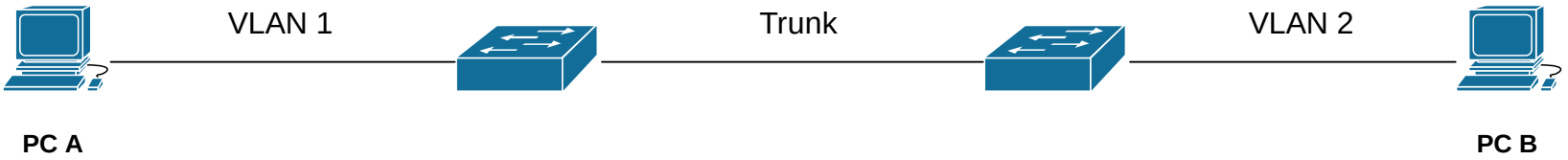
Src. MAC: A

Dst. MAC: B

Src. IP: A

Dst. IP: B

Double-tagging: Principle ③



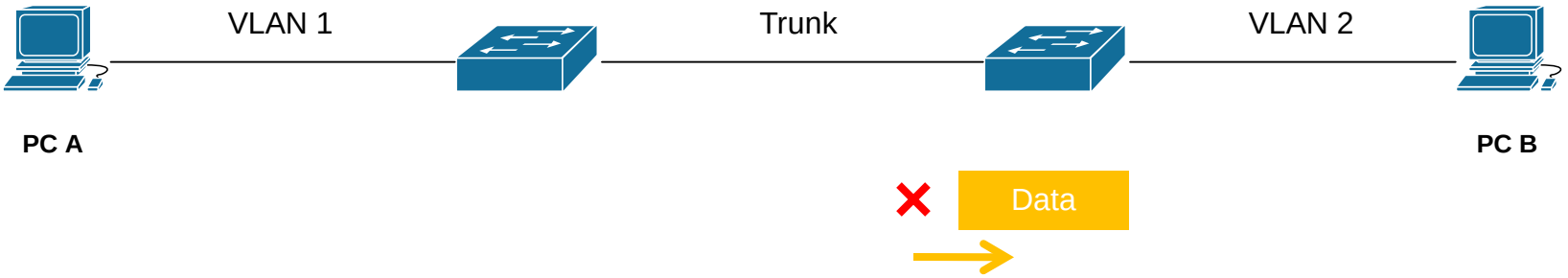
Src. MAC: A

Dst. MAC: B

Src. IP: A

Dst. IP: B

Double-tagging: Principle ④



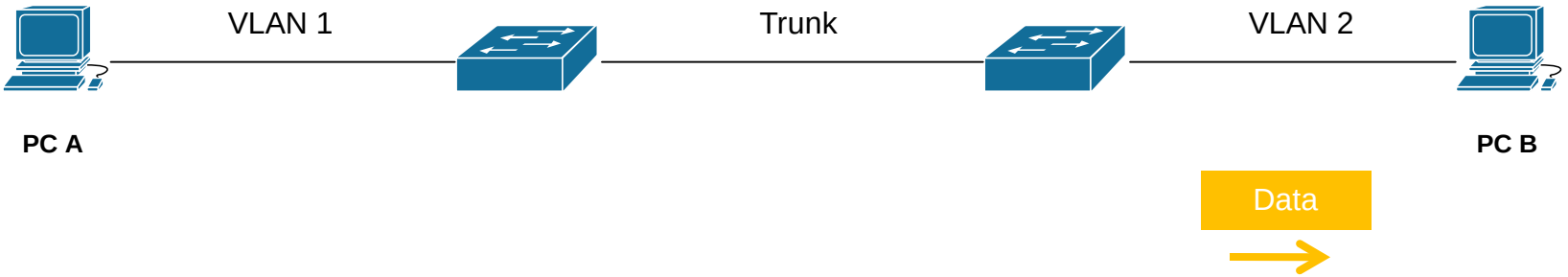
Src. MAC: A

Dst. MAC: B

Src. IP: A

Dst. IP: B

Double-tagging: Principle ⑤



Src. MAC: A

Dst. MAC: B

Src. IP: A

Dst. IP: B

Switch Spoofing Attack

Switch-spoofing

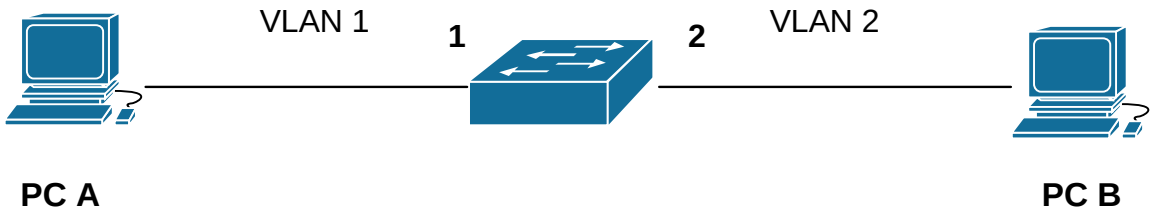
- Attacker pretends to be a switch and creates trunk
- ISL and 802.1Q compatible
- Works even in the single switch scenario
- Operational vs. Administrative mode

```
Switch(config-if)# switchport mode {access|trunk|dynamic}
```

  - Dynamic Auto: passive, default for 2960 and 3560
  - Dynamic Desirable: active, default for 2950 and 3550

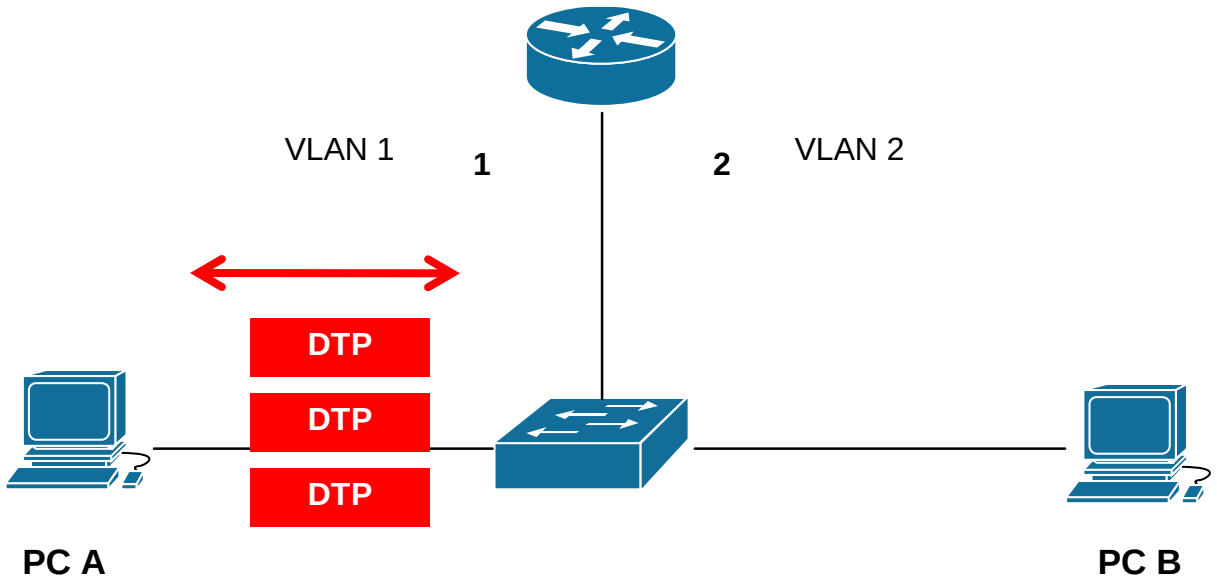
Switch-spoofing: Principle

Administrative Mode:	dynamic desirable
Operational Mode:	static access



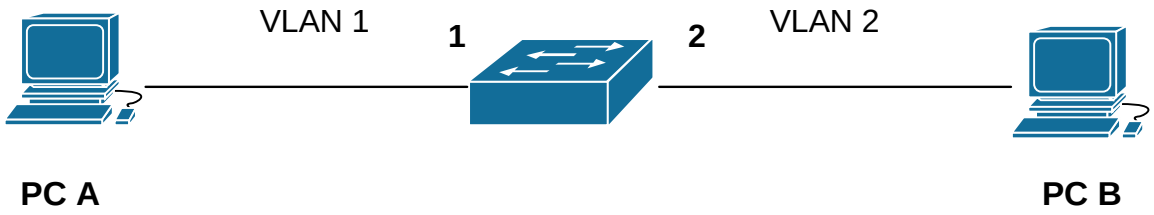
Switch-spoofing: Trunk Negotiation ①

```
01:05:57: DTP-queue:Fa0/1:Queuing DTP packet 01:05:57: DTP-
state:Fa0/1:Starting state transition from state S2:ACCESS, event
2b:PKT TO TRK ../dyntrk/dyntrk_fsm.c:631
```



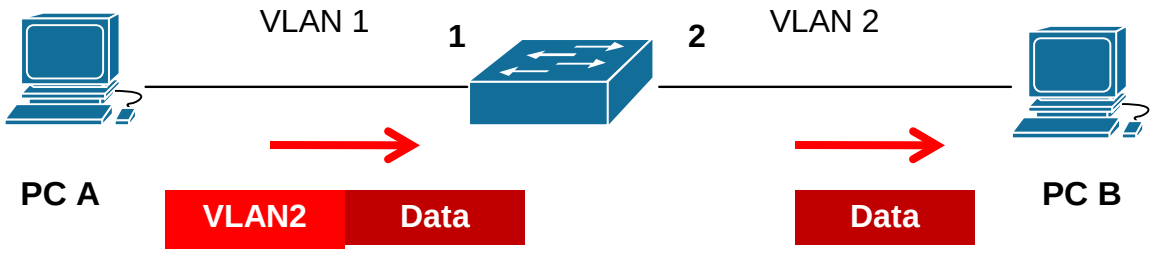
Switch-spoofing: Trunk Negotiation ②

Administrative Mode:	dynamic desirable
Operational Mode:	trunk



Switch-spoofing: Communication

Administrative Mode:	dynamic desirable
Operational Mode:	trunk



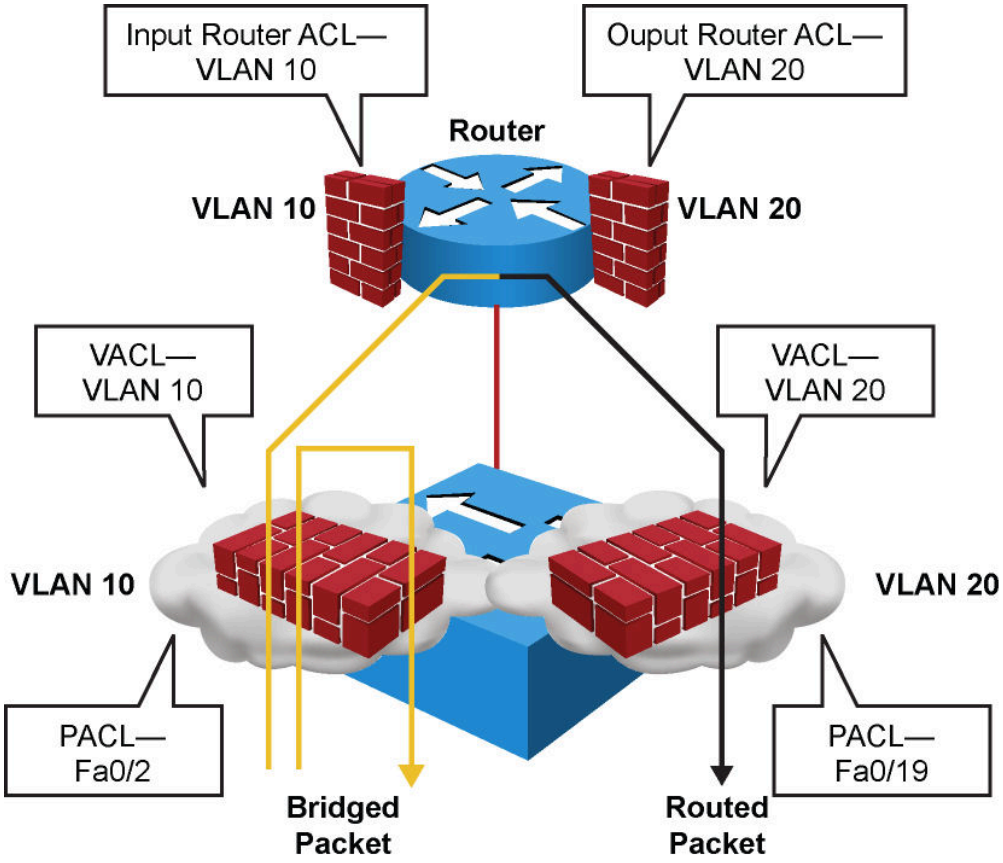
VLAN Attacks & Switch-spoofing Countermeaures

General Mitigation Steps

- Configure all unused ports as access ports so that trunking cannot be negotiated across those links (DTP is deactivated)
- Place all unused ports in the shutdown state and associate them with a parking-lot VLAN
- When establishing a trunk link, purposefully configure arguments to achieve the following results:
  - The native VLAN is different from any data VLANs
  - Trunking is **`switchport nonegotiate`** rather than negotiated
  - The specific VLAN range is carried on the trunk with **`switchport trunk allow vlan`**. This ensures that the native VLAN will be pruned along with any other VLANs not explicitly allowed on the trunk
- Allow tagging of native VLAN

Catalyst Multilayer Switch ACL Types

- **Router access control lists (RACL)**
  - Supported in the TCAM hardware on Cisco multilayer switches
  - In Catalyst switches, RACL can be applied to any routed interface
- **Port access control list (PACL)**
  - Filters traffic at the port level
  - PACL's can be applied on a Layer 2 switch port, trunk, or EtherChannel port
  - PACL's act at the Layer 2 port level but can filter based on Layer 3/Layer 4
- **VLAN access control lists (VACL)**
  - Apply to all traffic in a VLAN
  - VACL's support filtering based on Ethertype and MAC addresses
  - VACL's are order-sensitive, analogous to route maps
  - VACL's can control traffic flowing within the VLAN or control switched traffic, whereas RACL's control only routed traffic.



Configuring VACL's

1) Define a VLAN access map:

```
Switch(config-if)# vlan access-map map_name [seq#]
```

2) Configure a match clause:

```
Switch(config-access-map)# match {drop [log]} | {forward [capture]}
| {redirect {{fastethernet | gigabitethernet | tengigabitethernet}
slot/port} | {port-channel channel_id}}
```

3) Configure an action clause:

```
Switch(config-access-map)# action {drop [log]} | {forward [capture]}
| {redirect {{fastethernet | gigabitethernet | tengigabitethernet}
slot/port} | {port-channel channel_id}}
```

4) Apply a map to VLANs:

```
Switch(config-if)# vlan filter map_name vlan_list list
```

5) Verify the VACL configuration:

```
Switch# show vlan access-map map_name
Switch# show vlan filter [access-map map_name|vlan vlan_id]
```

Example: VACL's ①

- Here a VACL is configured to drop all traffic from network 10.1.9.0/24 on VLAN 10 and 20 and drop all traffic to Backup Server 0000.1111.4444.

```
switch(config)# access-list 100 permit ip 10.1.9.0 0.0.0.255 any
switch(config)# mac access-list extended BACKUP_SERVER
switch(config-ext-mac)# permit any host 0000.1111.4444
switch(config)# vlan access-map XYZ 10
switch(config-map)# match ip address 100
switch(config-map)# action drop
switch(config-map)# vlan access-map XYZ 20
switch(config-map)# match mac address BACKUP_SERVER
switch(config-map)# action drop
switch(config-map)# vlan access-map XYZ 30
switch(config-map)# action forward
switch(config)# vlan filter XYZ vlan-list 10,20
```

Example: VACL's ②

- VACL evaluation is dependent on block order
- VACL may contain both match IP and match MAC statements

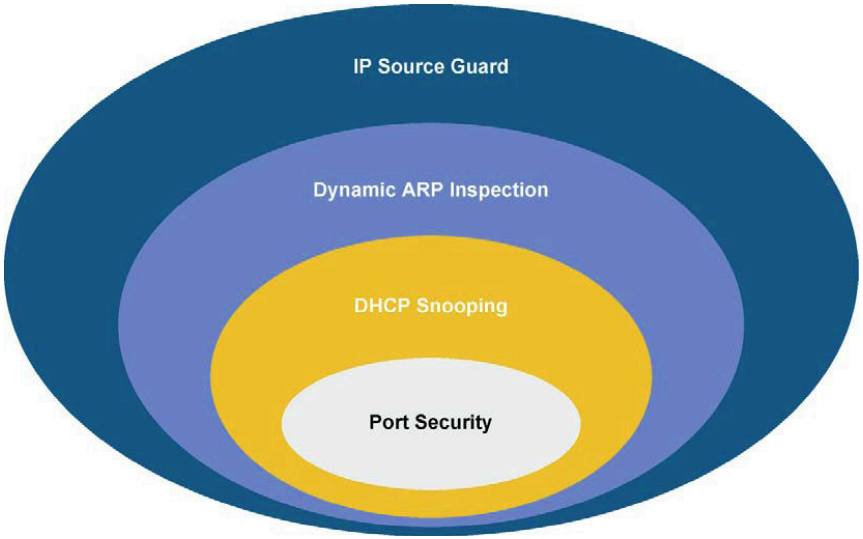
```
ip access-list standard Machine100
 permit 192.0.2.100

ip access-list extended WinServ
 permit tcp any any range 135 139
 permit udp any any range 135 139
 permit tcp any any eq 445
 permit udp any any eq 445
 permit udp any any eq 1900

mac access-list extended IPX
 permit any any 0x8137 0x0
 permit any any lsap 0xFFFF
 permit any any lsap 0xE0E0
```

```
vlan access-map V50 10
 match ip address Machine100
 action forward

vlan access-map V50 20
 match ip address WinServ
 match mac address IPX
 action drop

vlan access-map V50 30
 action forward

vlan filter V50 vlan-list 50
```

Identity Spoofing Attacks

Spoofing

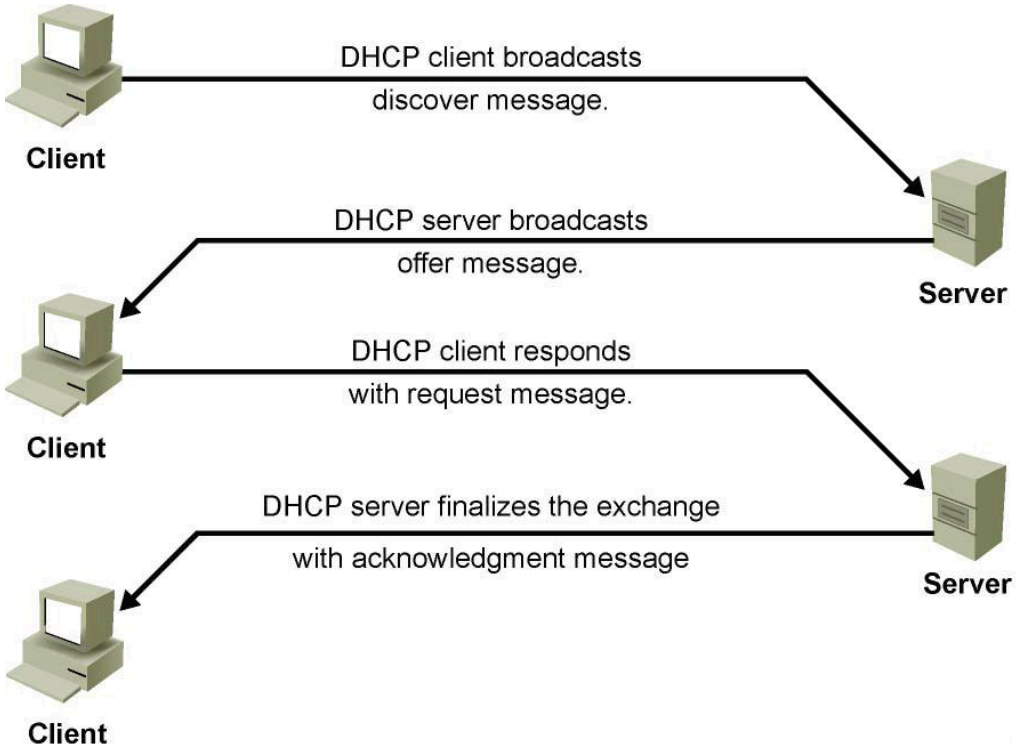
- **Spoofing** is attempt to fake device's identity
- Three types do exist according to used protocol
  - DHCP Spoofing
  - ARP Spoofing
  - IP Spoofing
- Catalyst switches contain several counter-measures



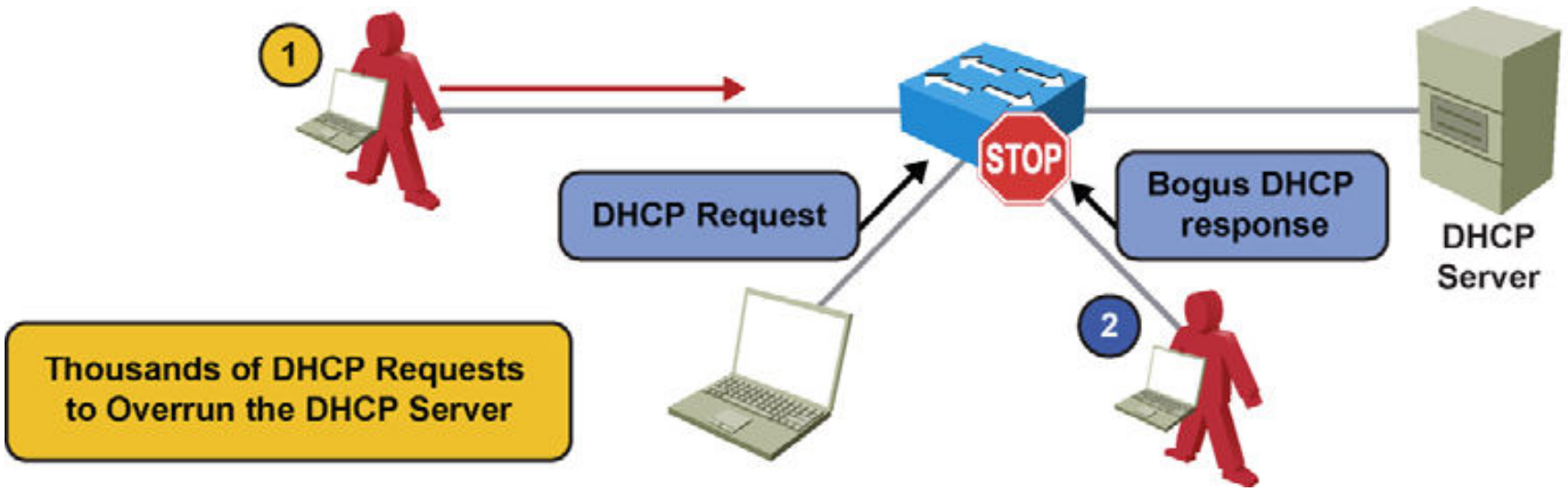
DHCP Spoofing Attack

DHCP Spoofing Attack

- Attacker can gain access to network traffic by spoofing DHCP responses
- The DHCP spoofing device replies to client DHCP requests. The legitimate server can reply also, but if the spoofing device is on the same segment as the client, its reply to the client might arrive first.
- The intruder's DHCP reply offers an IP address and supporting information that designates the intruder as the default gateway or DNS server.



DHCP Spoofing Attacks

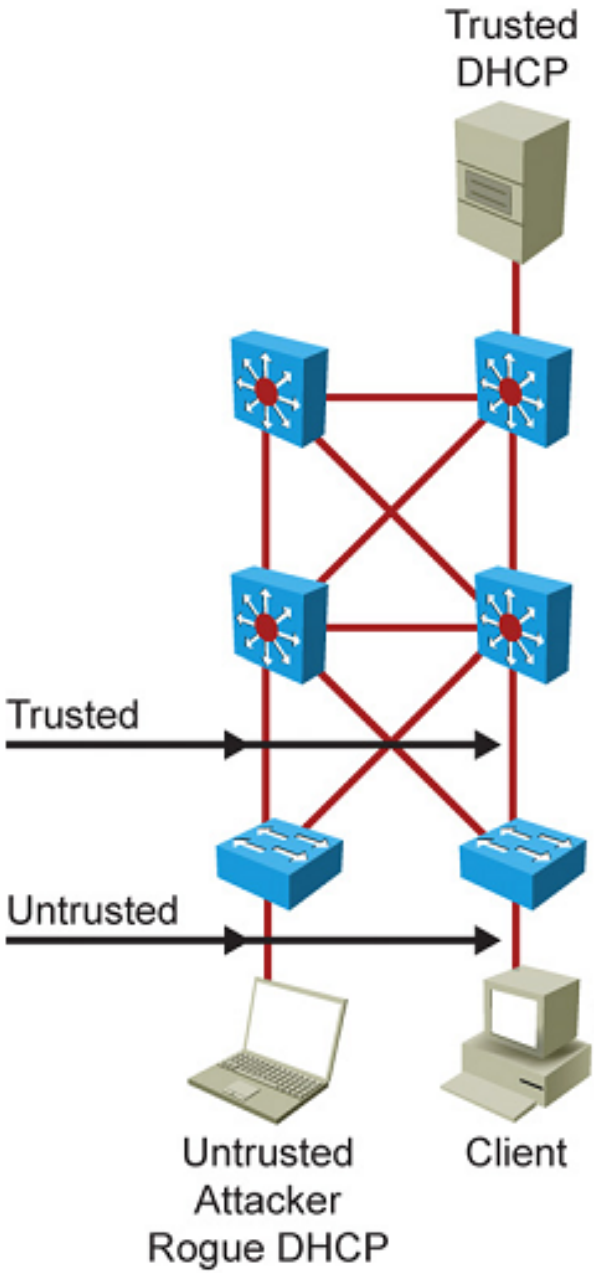


- **Scenario 1**
  - An attacker launches a DoS attack by sending thousands of DHCP requests
  - The DHCP server does not have the capability to determine whether the request is genuine and therefore might end up exhausting all the available IP addresses
  - This results in a legitimate client not getting a IP address via DHCP
- **Scenario 2**
  - Attacker attaches a DHCP server to the network and has it assume the role of the DHCP server for that segment
  - This enables the intruder to give out false DHCP information for the default gateway and domain name servers, which points clients to the hacker's machine

DHCP Snooping

DHCP Snooping

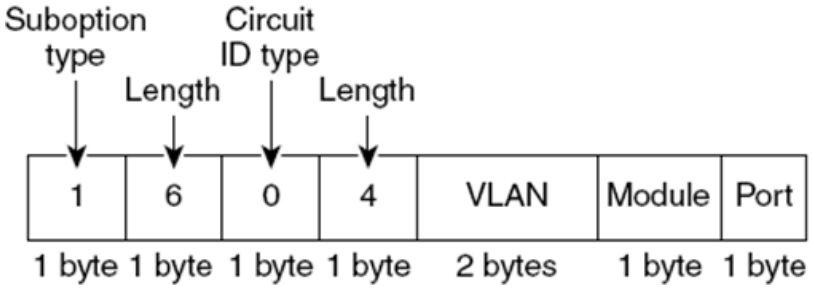
- **DHCP Snooping** prevents client attacks on the DHCP server and switch
  - DHCP requests (discover) and responses (offer) are tracked
  - Rate-limiting requests on untrusted interfaces limit DoS attacks on DHCP server (transition to err-disabled state)
  - Deny responses on untrusted interfaces in order to stop malicious or errant DHCP servers
- Ports differentiation
  - On untrusted ports are connected end-stations
  - On trusted ports or behind them are DHCP servers
  - Default port type is untrusted
- **DHCP Snooping Database**
  - Source MAC, leased IP address, lease time, VLAN and interface
  - It is used also by DAI and IPSG



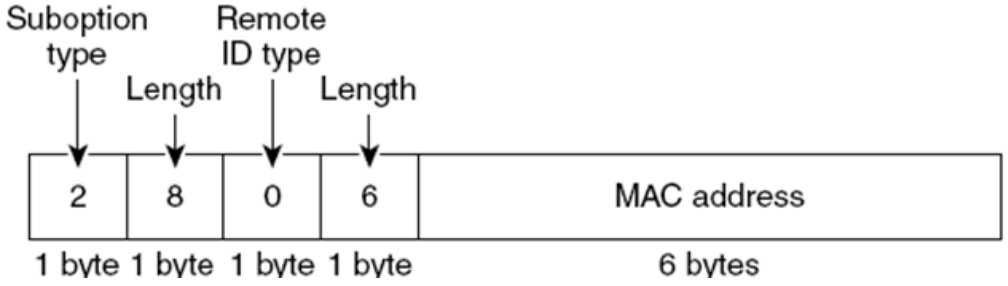
Option-82

- `IF` DHCP Snooping intercepts request `THEN` it inserts Option-82 into the message
- **Option-82** is two parts information field containing switch and its interface on which request was received
  - Circuit ID identifies clients interface
  - Remote ID identifies switch to which client is attached

Circuit ID Suboption Frame Format



Remote ID Suboption Frame Format



Operation Principle

- DHCP Snooping discards…
  i. server DHCP messages (Offer, Ack, Nak, Leasequery) received on untrusted port
  ii. client DHCP messages (Discover, Request) containing different chaddr MAC address than what is source MAC of sender
  iii. client DHCP Release and Decline messages with MAC addresses that are present on different interface in DHCP Snooping Database
  iv. all messages received on untrusted port containing different relay agent address than 0.0.0.0 or contains Option-82
- DHCP Snooping forwards…
  - client DHCP messages only to trusted ports
  - server DHCP messages to recipient based on Option-82 information (even thou they are broadcasts or are intended to unknown receiver)

Configuration

1) Enable DHCP globally

```
Switch(config)# ip dhcp snooping
```

2) Enable DHCP Option 82

```
Switch(config)# ip dhcp snooping information option
```

3) Configure DHCP server interfaces or uplink ports as trusted

```
Switch(config-if)# ip dhcp snooping trust
```

4) Configure the number of DHCP packets per second (pps)

```
Switch(config-if)# ip dhcp snooping limit rate rate
```

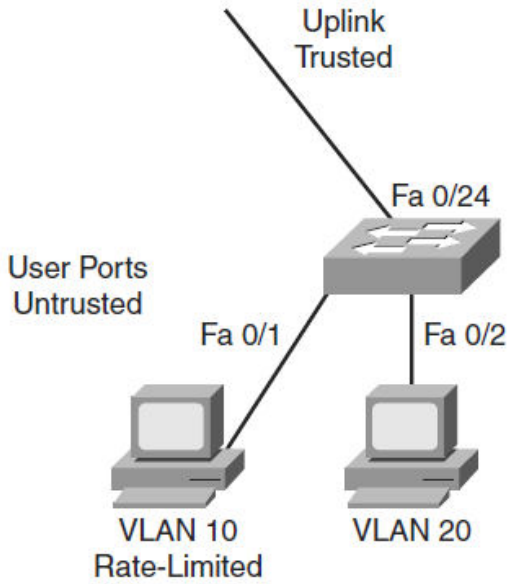
5) Enable DHCP snooping on specific VLANs

```
Switch(config)# ip dhcp snooping vlan number [number]
```

6) Verify the configuration

```
Switch# show ip dhcp snooping
```

Example




```
switch(config)# ip dhcp snooping
switch(config)# ip dhcp snooping information option
switch(config)# ip dhcp snooping vlan 10,20
switch(config)# interface fastethernet 0/1
switch(config-if)# description Access Port
switch(config-if)# ip dhcp limit rate 5
switch(config)# interface fastethernet 0/24
switch(config-if)# description Uplink
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 10,20
switch(config-if)# ip dhcp snooping trust
```

Verification

```
switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10,20
DHCP snooping is operational on following VLANs:
10,20
DHCP snooping is configured on the following L3 Interfaces:
Insertion of option 82 is enabled
circuit-id default format: vlan-mod-port
remote-id: 001a.e372.ab00 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface          Trusted  Allow option     Rate limit (pps)
----------------   -------  -------------    --------------
FastEthernet0/1    no           no                   5
FastEthernet0/24   yes          yes              unlimited
```

Option-82 Troubles ①

- IF trusted DHCP server is in different than VLAN clients THEN everything will work fine
- IF trusted DHCP server is on same VLAN THEN
  - DHCP server stops assigning addresses and debugs contains

```
Router# debug ip dhcp server packet
*Sep 9 01:59:40: DHCPD: inconsistent relay information.
*Sep 9 01:59:40: DHCPD: relay information option exists, but giaddr is zero
```

  - It is because Option-82 is empty
  - The solution is to allow DHCP server to accept this kind of DHCP messages either globally or per interface

```
! Globally
Router(config)# ip dhcp relay information trust-all
! Per-interface
Router(config)# int fa0/1
Router(config-if)# ip dhcp relay information trusted
```

Option-82 Troubles ②

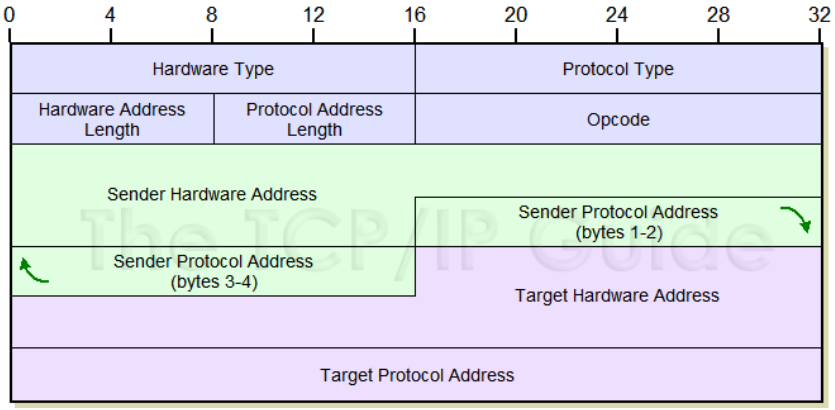
- In some situations access switch is able to insert Option-82 but is unable to perform DHCP Snooping
  - DHCP Snooping is then delegated to upstream switch
- The problem is which port type to use on upstream switch
  - `IF` it is untrusted `THEN` DHCP Snooping discards all messages (according to rule iv)
  - `IF` it is trusted `THEN` DHCP Snooping will not be updated by those messages
- This is solved on upstream switch by following commands:

```
! Globally
AggSw(config)# ip dhcp snooping option allow-untrusted
! Per-interface
AggSw(config)# int fa0/1
AggSw(config-if)# ip dhcp snooping option allow-untrusted
```

ARP Spoofing Attack

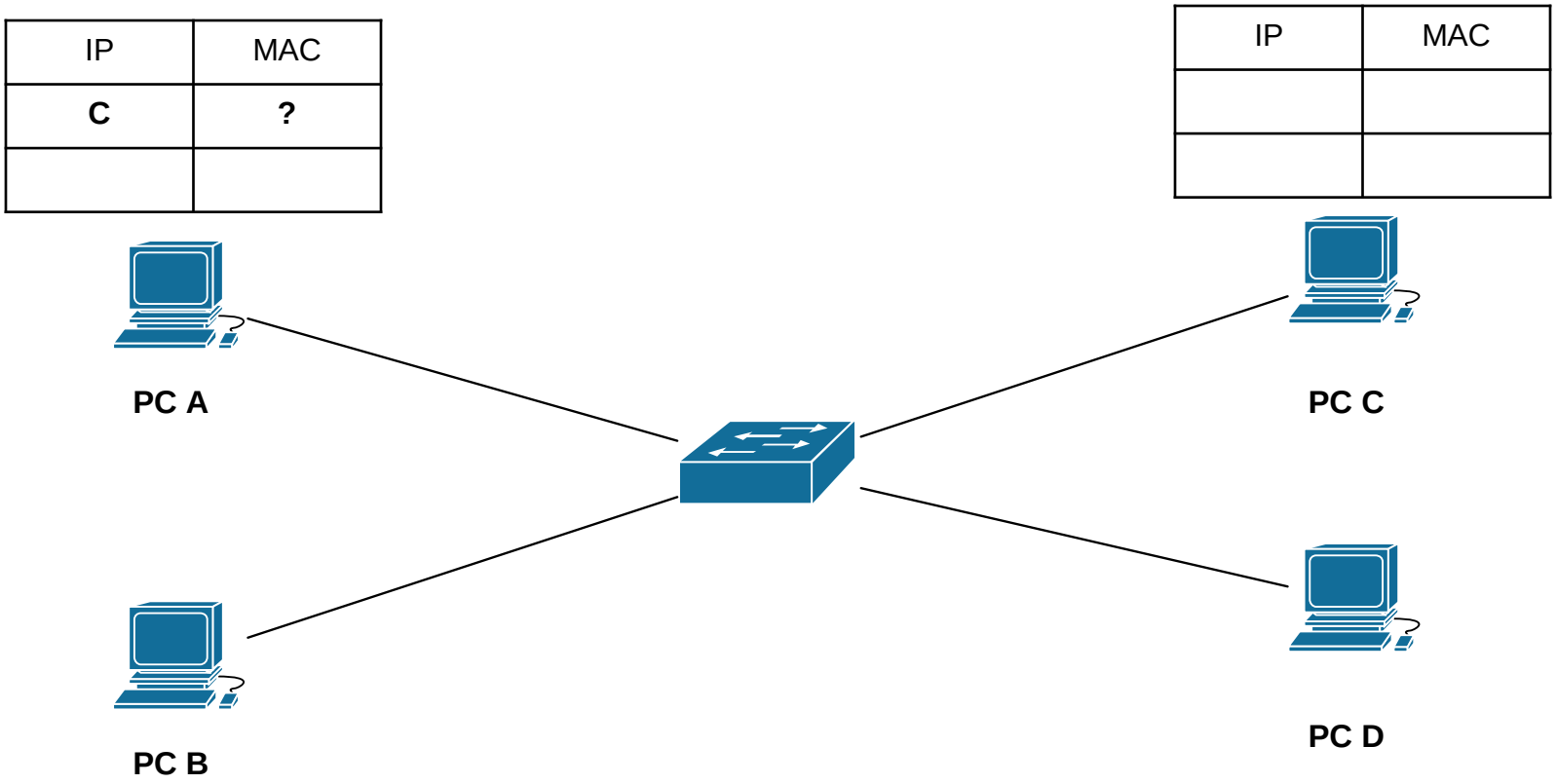
ARP Spoofing

- ARP Spoofing sends unsolicited so called gratuitous ARP response message in order to map different MAC address to a given IP address
- Two possible outcomes
  - Denial of Service by mapping non-existing MAC to IP
  - Man-in-the-Middle by mapping attackers MAC to IP



- *Tools*
  - Arpspoof
  - Ettercap
  - Cain & Abel

MAC Resolving: Initiation



MAC Resolving: Request ①

IP	MAC
C	**?**

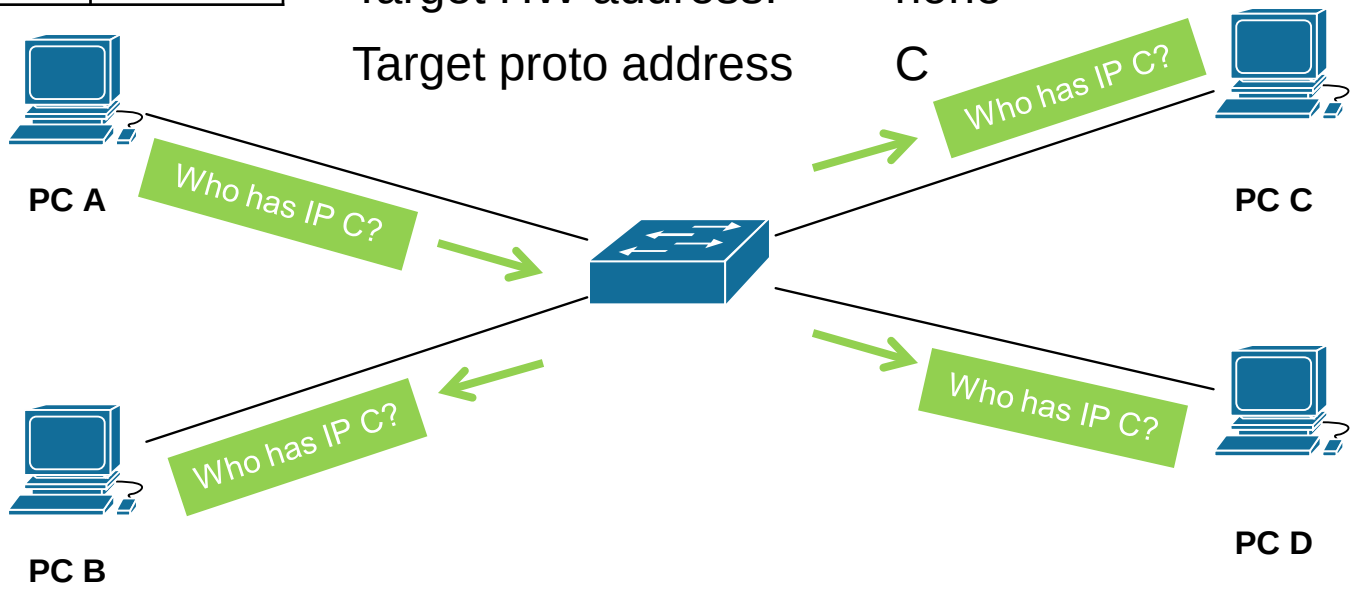
Sender HW addres: A

Sender proto address: A

Target HW address: none

Target proto address C

IP	MAC



MAC Resolving: Request ②

IP	MAC
C	?

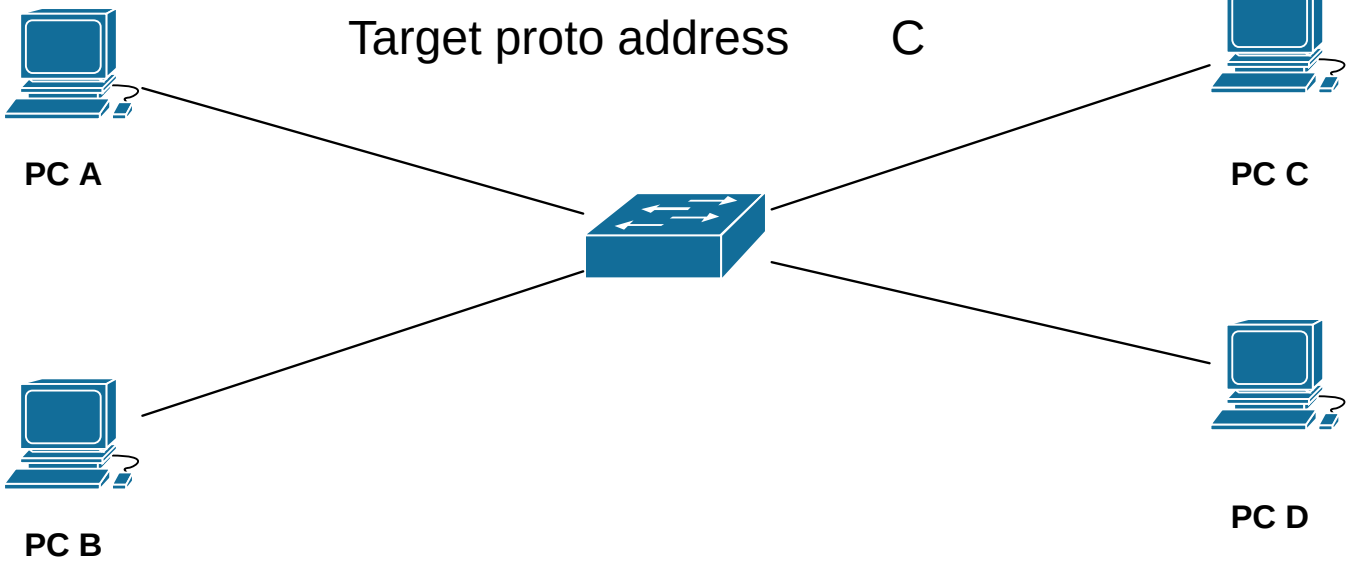
Sender HW addres: A

Sender proto address: A

Target HW address: none

Target proto address C

IP	MAC
A	A



MAC Resolving: Reply ①

IP	MAC
C	**?**

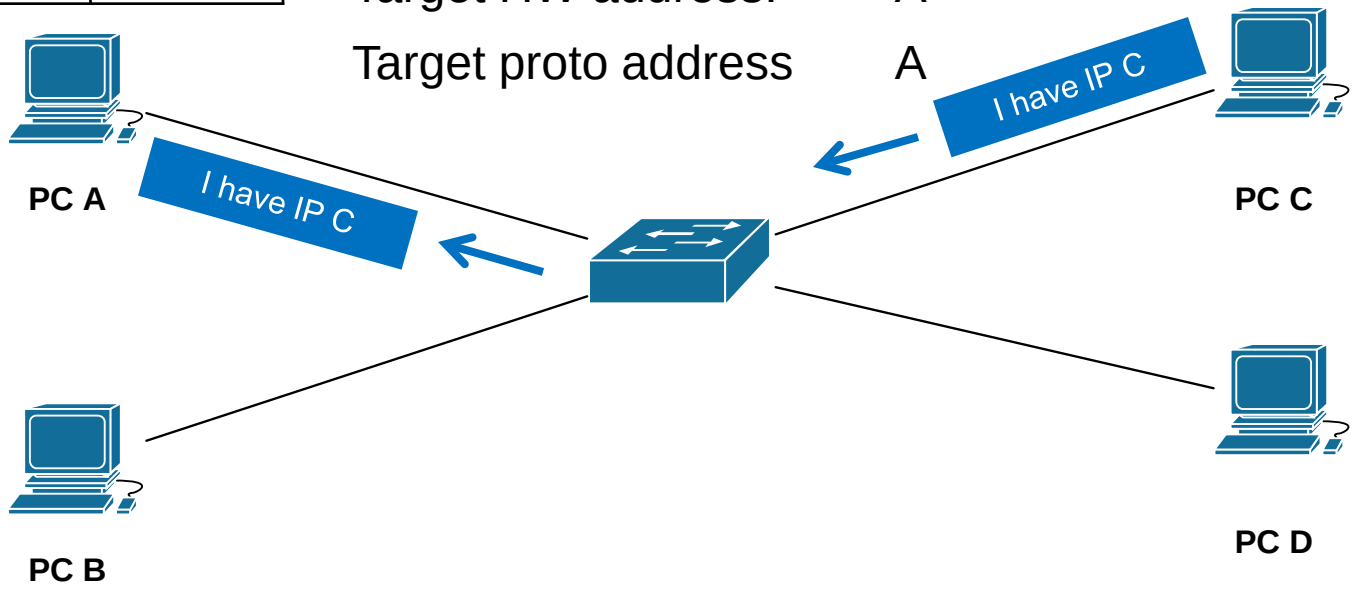
Sender HW addres: C

Sender proto address: C

Target HW address: A

Target proto address A

IP	MAC
A	**A**



MAC Resolving: Reply ②

IP	MAC
C	**C**

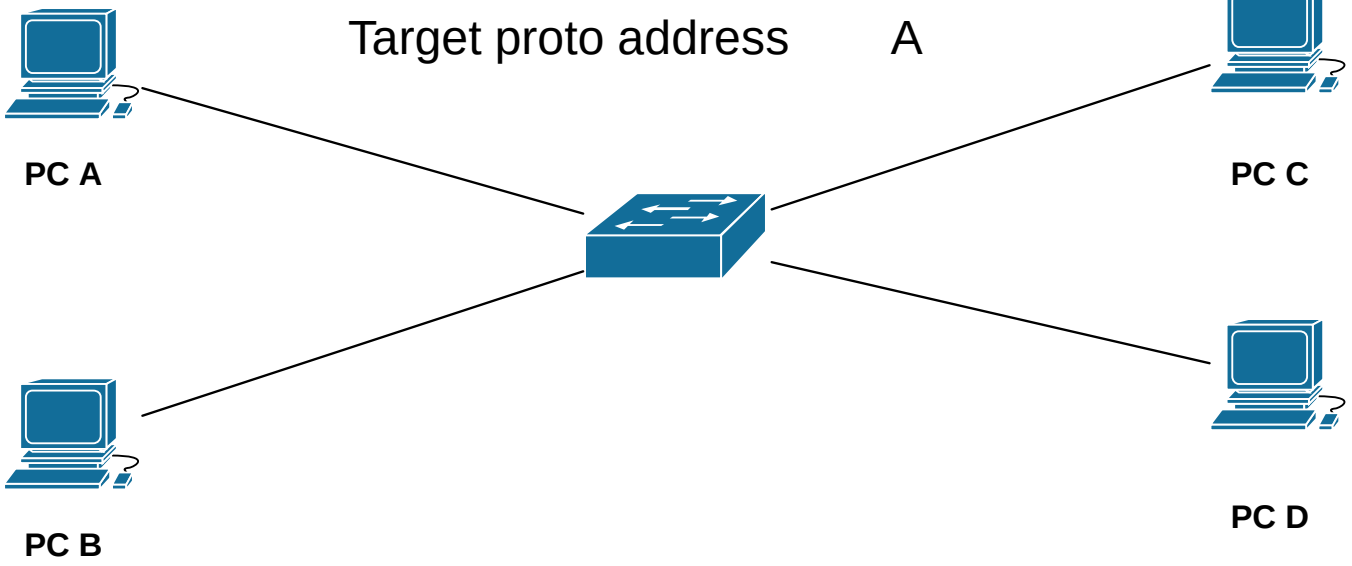
Sender HW addres: C

Sender proto address: C

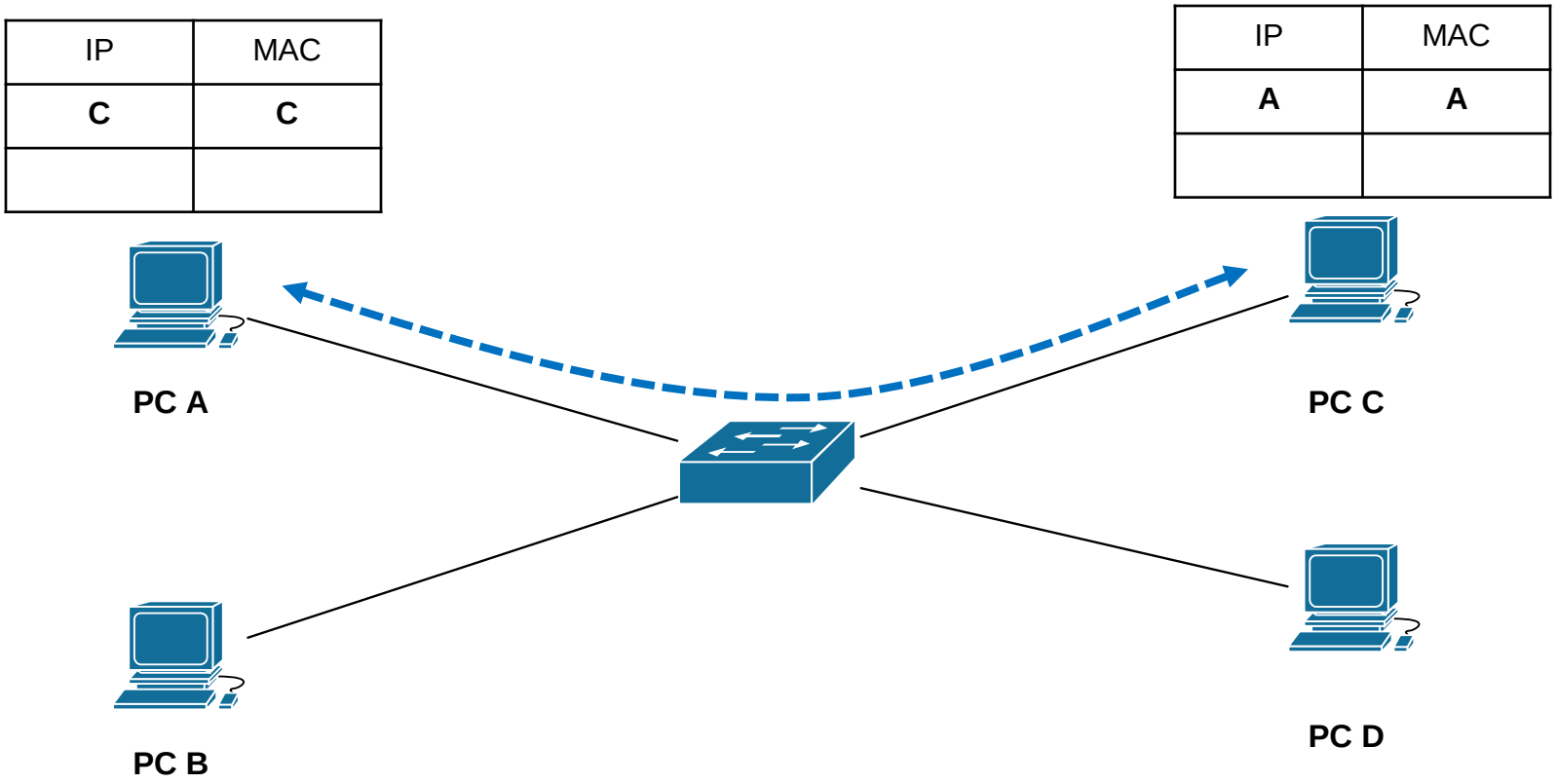
Target HW address: A

Target proto address A

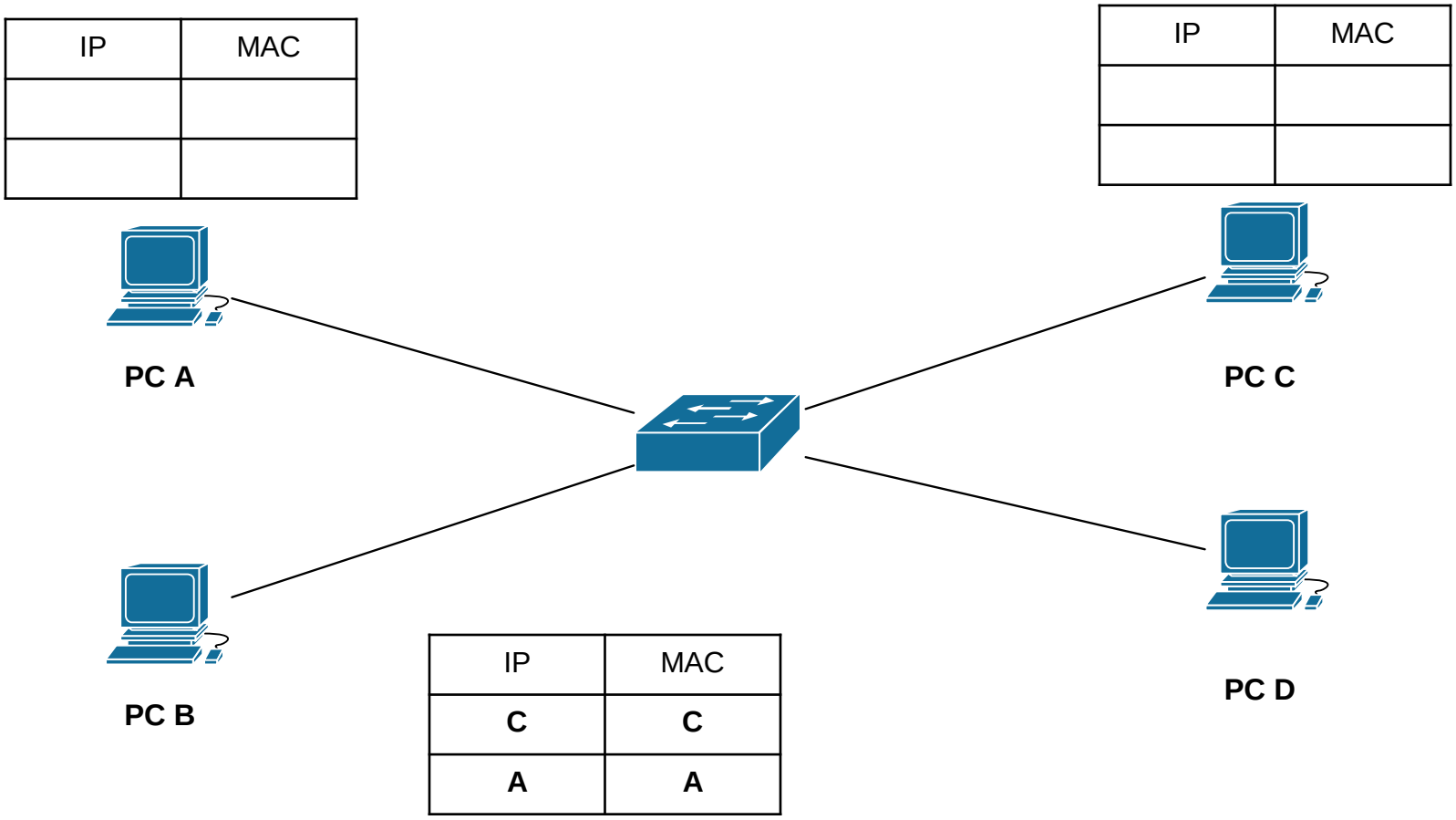
IP	MAC
A	**A**



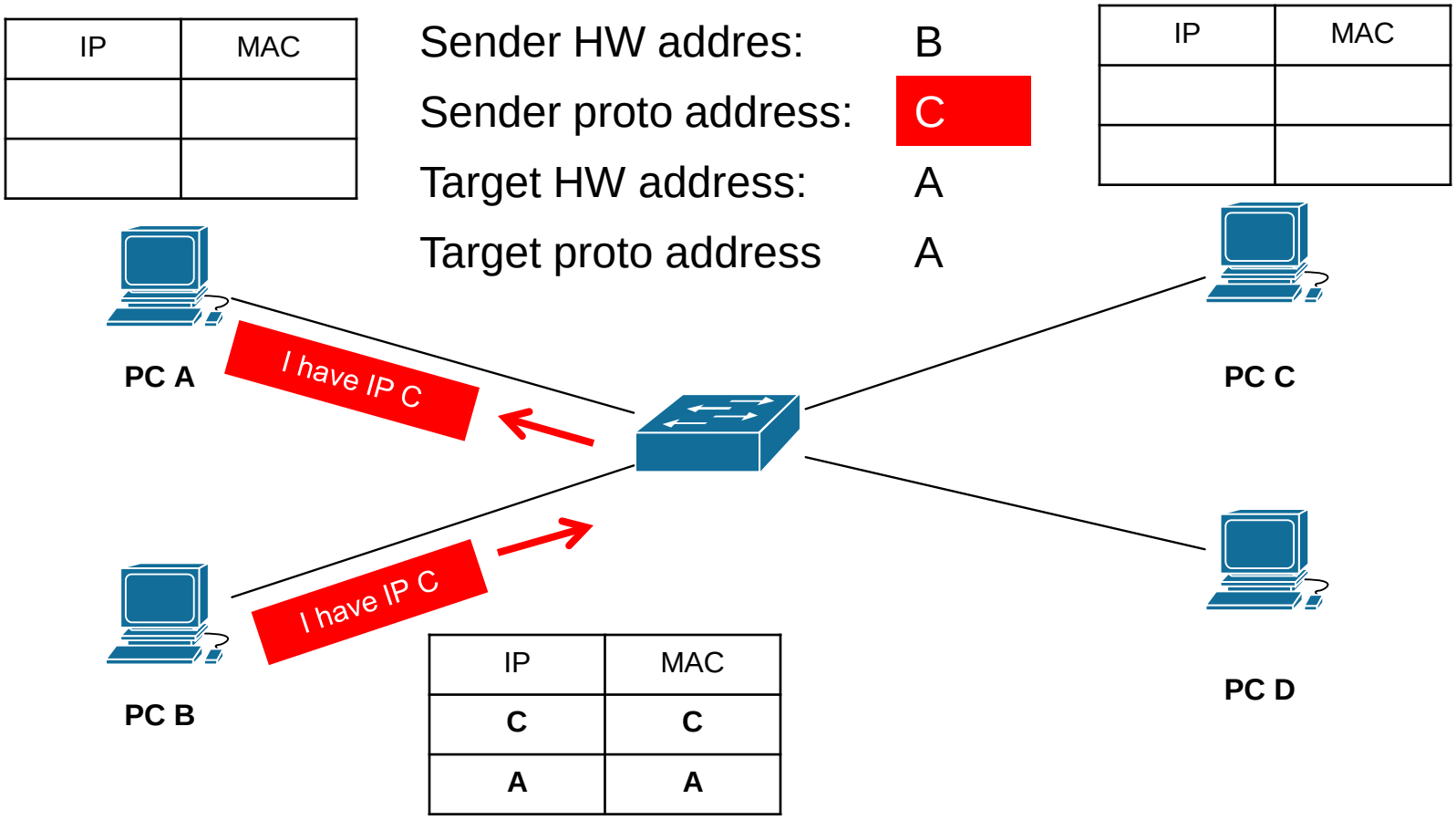
MAC Resolving: Conclusion



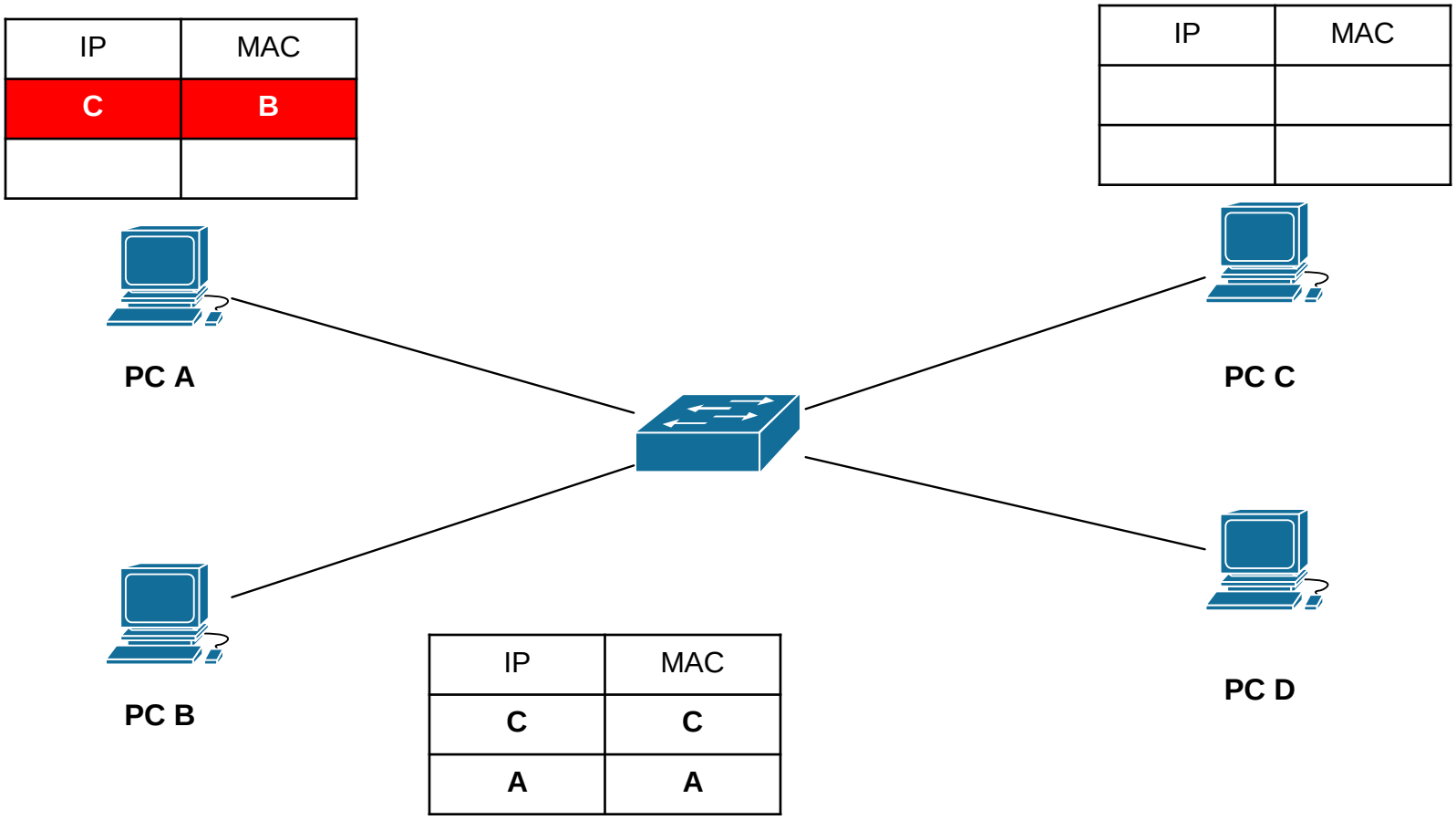
ARP MitM



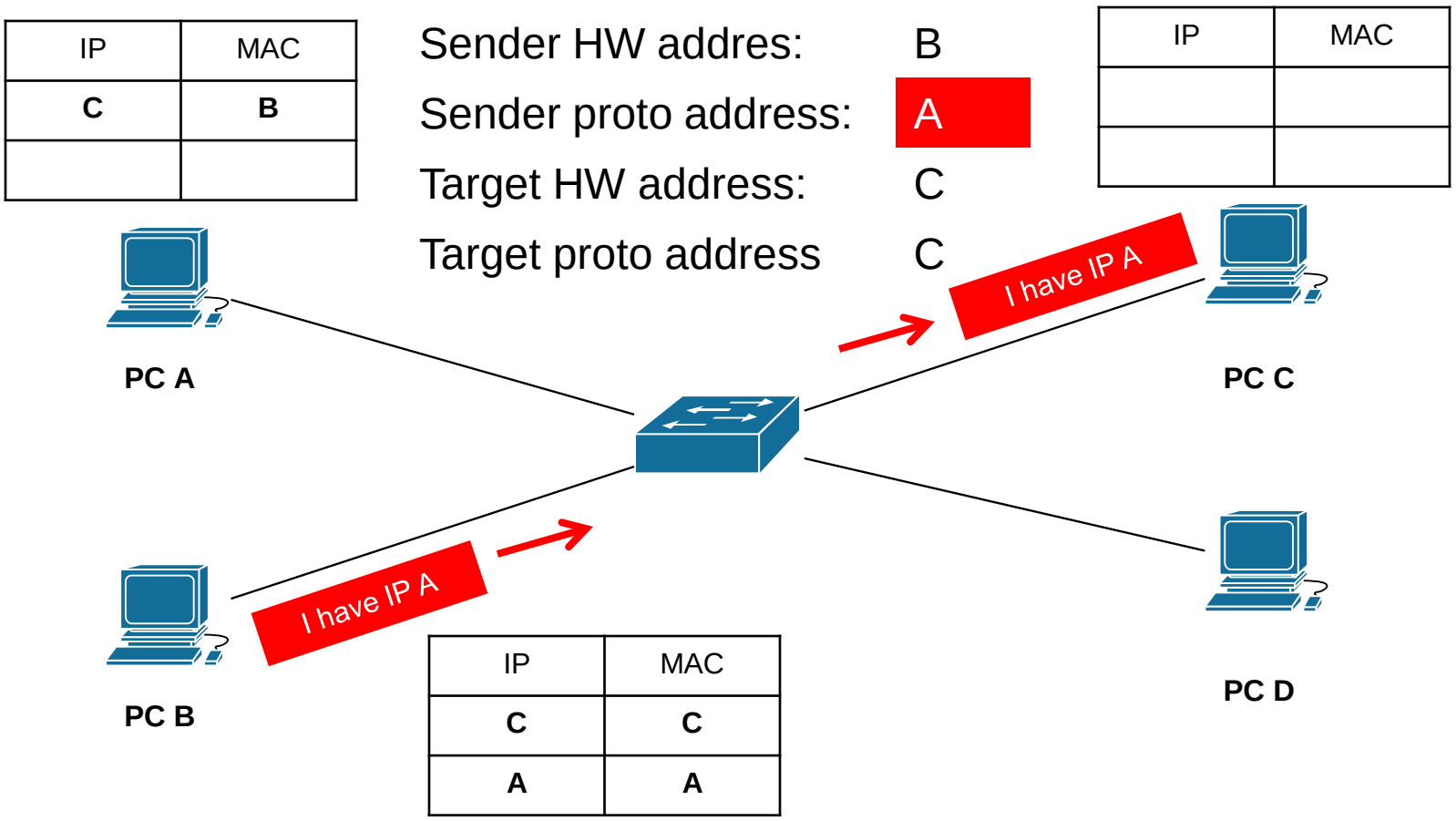
ARP MitM: Cache Poisoning ①



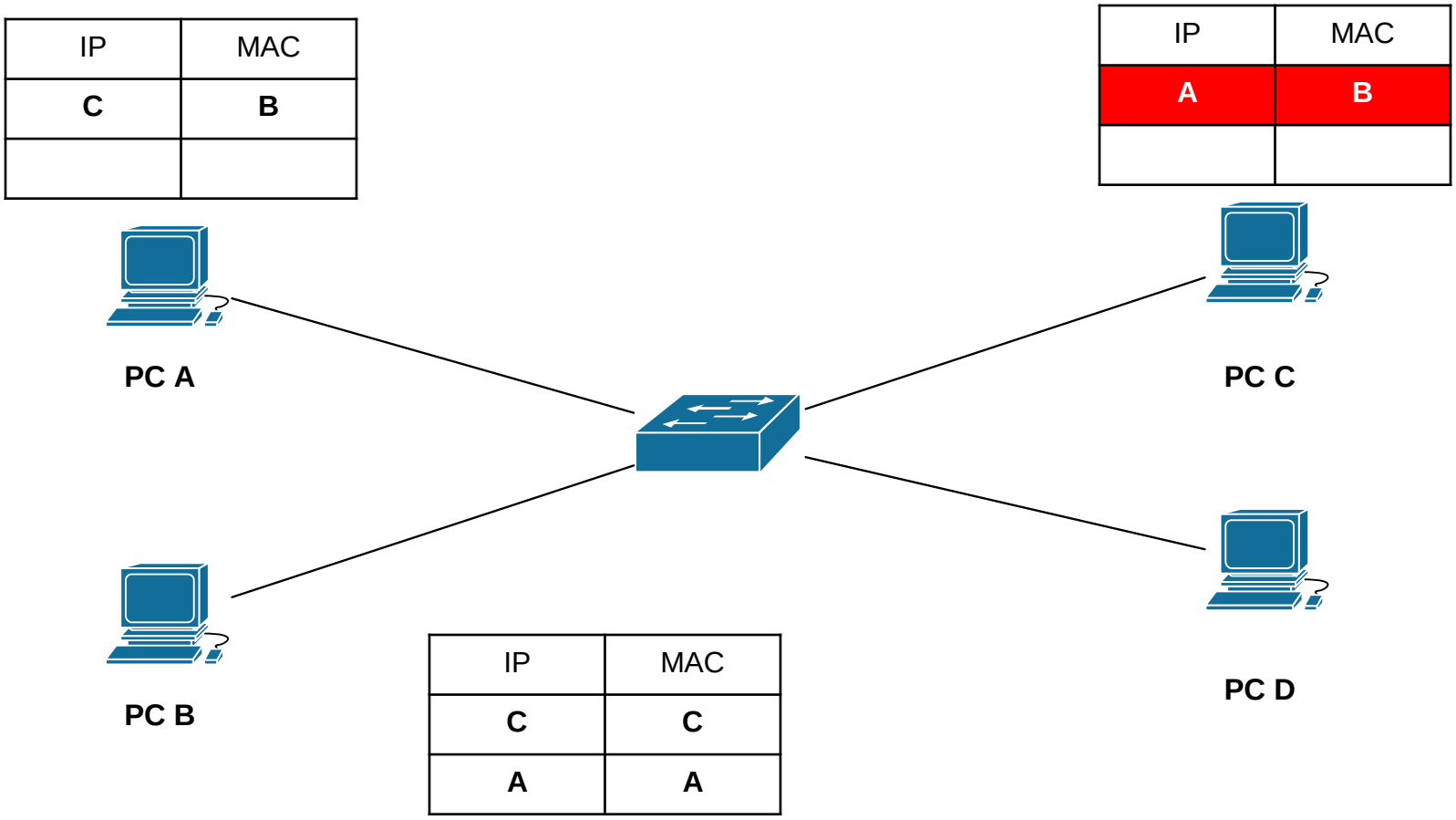
ARP MitM: Cache Poisoning ②



ARP MitM: Cache Poisoning ③



ARP MitM: Cache Poisoning ④



ARP MitM: Forwarding ①

IP	MAC
C	**B**

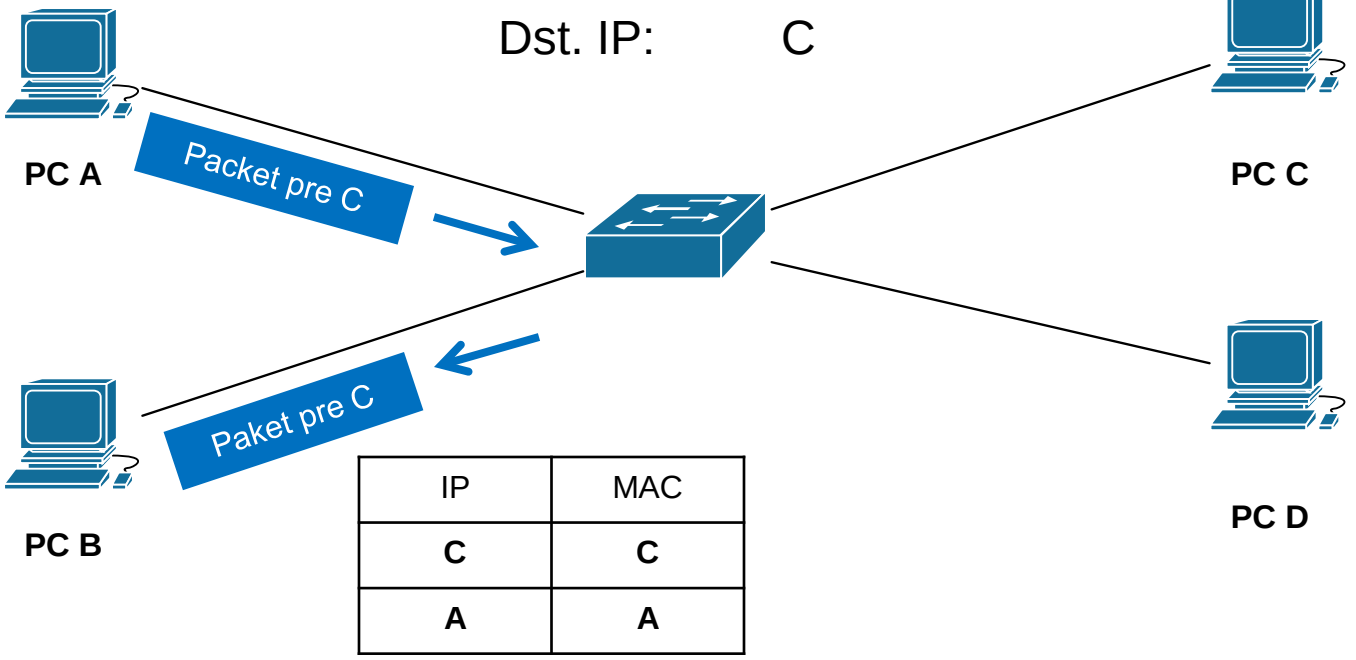
Src. MAC: A

Dst. MAC: B

Src. IP: A

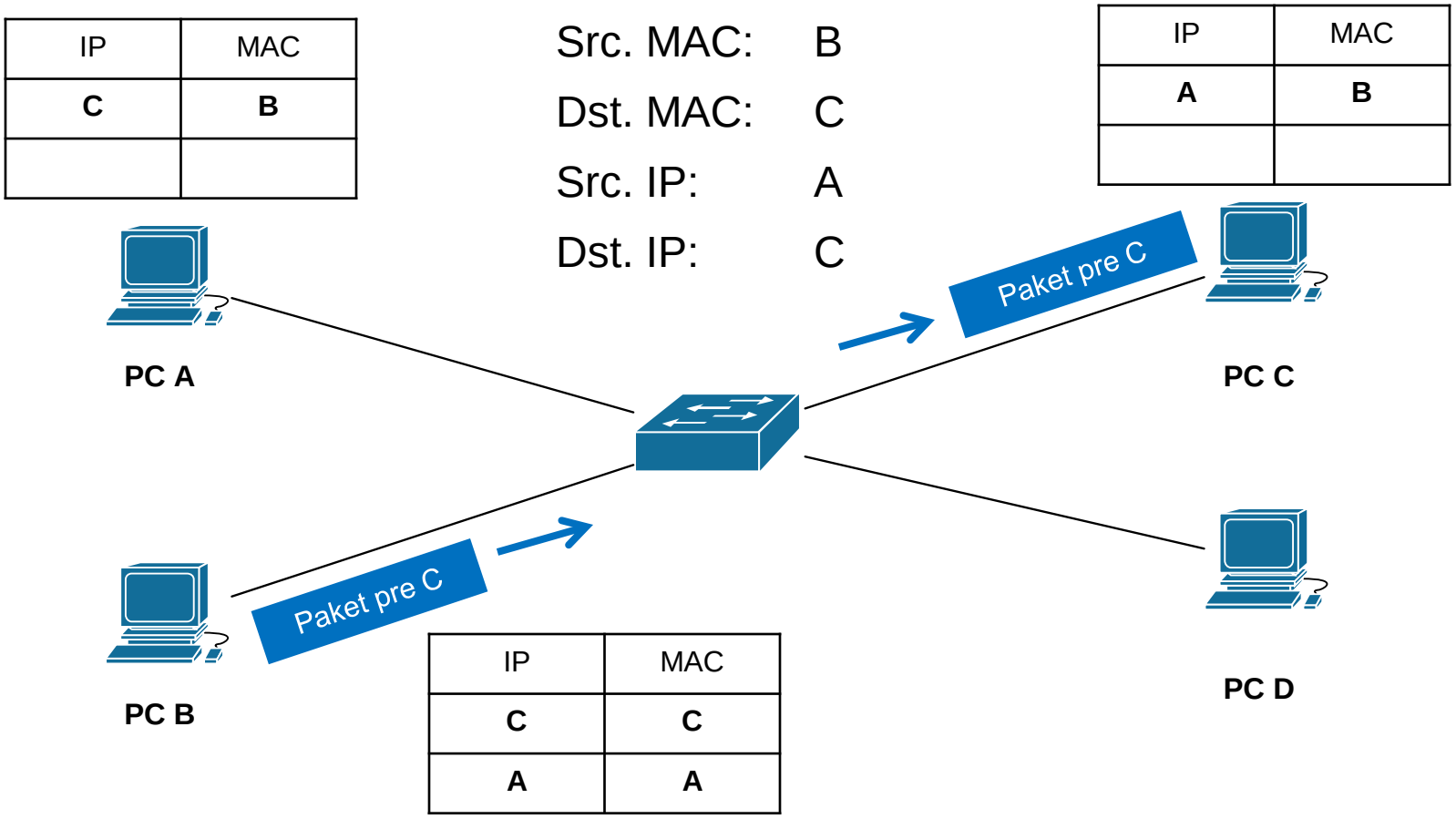
Dst. IP: C

IP	MAC
A	**B**

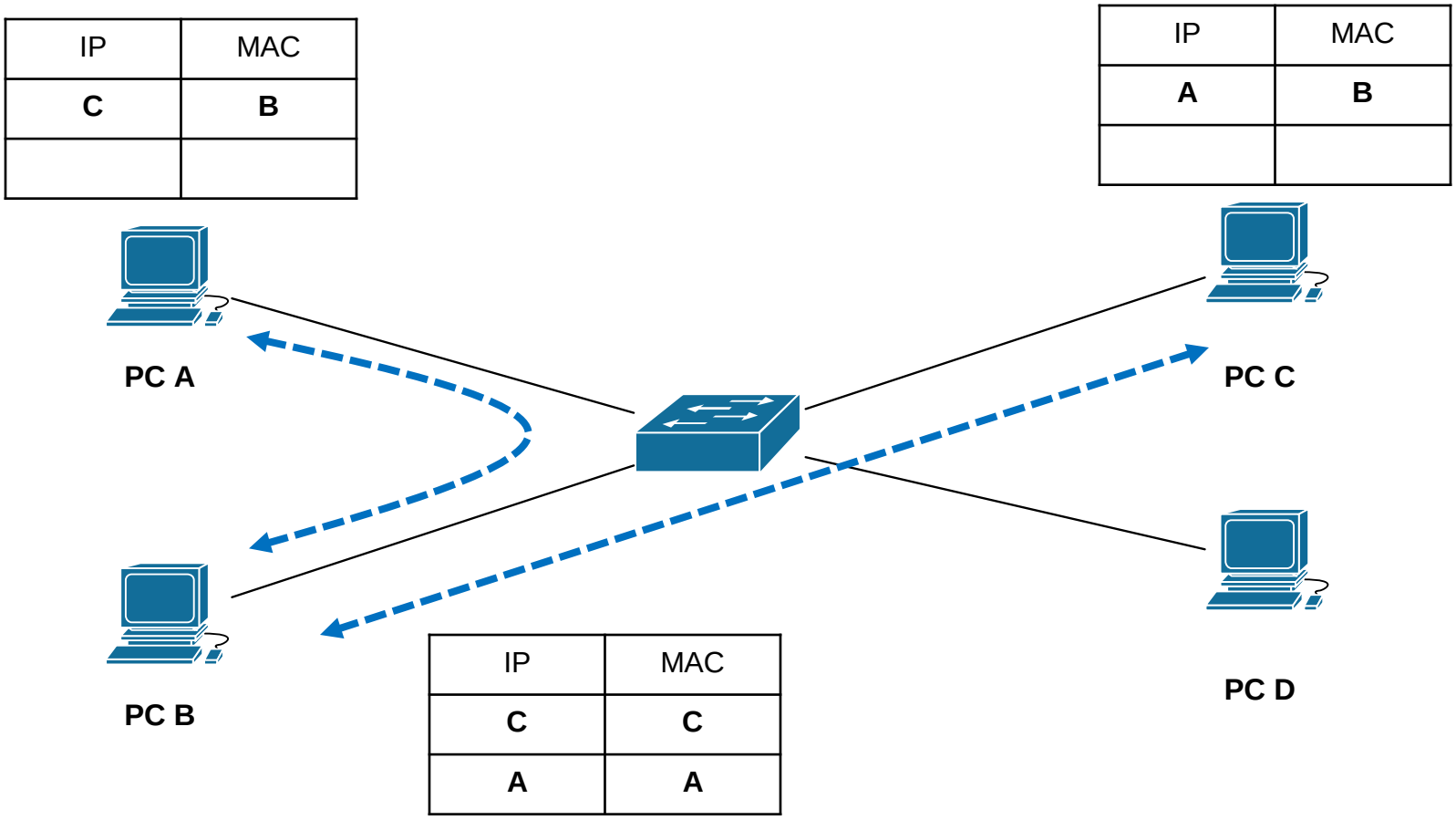


IP	MAC
C	**C**
A	**A**

ARP MitM: Forwarding ②



ARP MitM: Forwarding ③



Client's ARP Cache

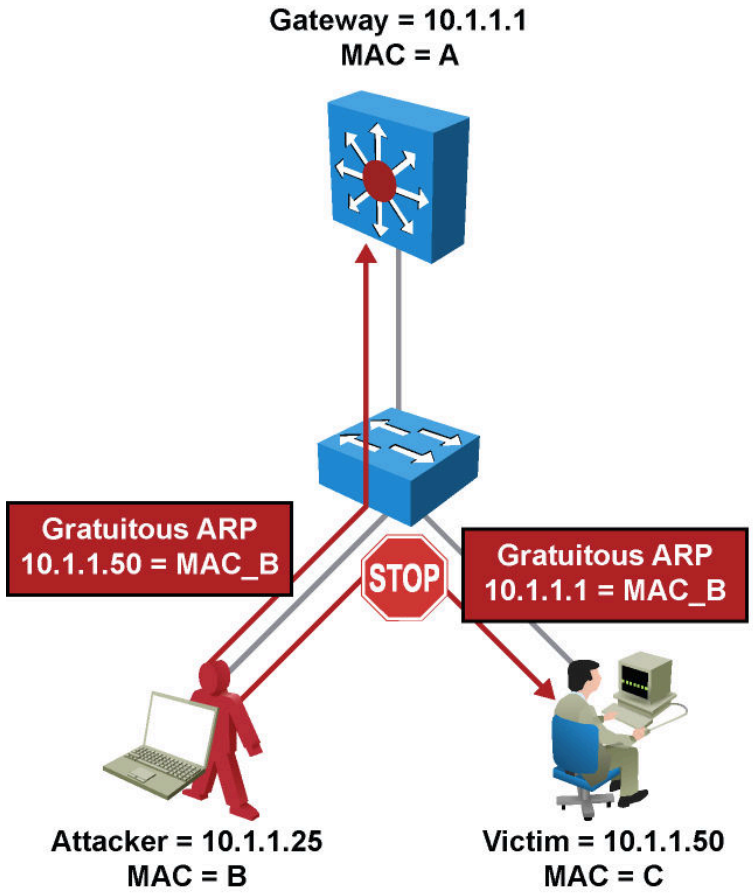
- Updated by ARP Response and ARP Request that should be answered
- Attack works only until timeout expires the record
- Each record has dynamic timout

Operating System	Default Timeout
Windows 2000/XP	10 minutes
Windows 2003 Server	10 minutes
Windows Vista	15-45 seconds
Windows 7	15-45 seconds
Cisco IOS 12.0	4 hours
Ubuntu	1 minute
FreeBSD	20 minutes

DHCP Spoofing Attack

Dynamic ARP Inspection

- **Dynamic ARP Inspection (DAI)** is precaution against ARP Spoofing attacks that uses DHCP Snooping Database to validate senders and receivers MAC and IP addresses
- Port types
  - Untrusted ports heading to end-devices
  - Trusted ports heading to other routers and switches
  - Default port type is untrusted
- DAI violation drops packet and logs incident
- DAI allows rate-limiting of ARP messages
  - By default it is set to 15 messages per second on untrusted ports



Configuration

1) Enable DAI on a VLAN or range of VLANs

```
Switch(config)# ip arp inspection ip arp inspection vlan vlan-range
```

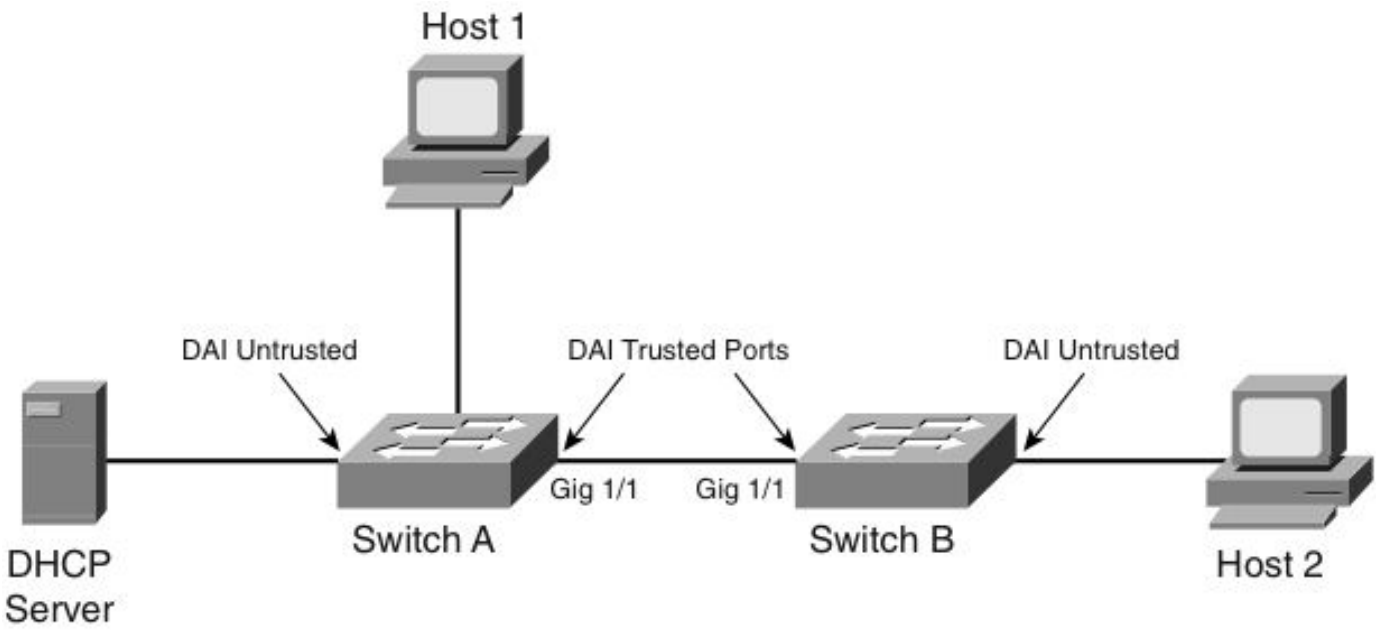
2) Enable DAI on an interface and sets the interface as a trusted interface

```
Switch(config-if)# ip arp inspection trust
```

3) Optionally, additional checks for src/dst MACs, IP different than 0.0.0.0/255.255.255.255 or multicast

```
Switch(config)#
ip arp inspection validate {[src-mac] [dst-mac] [ip [allow-zeros] ]}
```

Example: DAI




```
SwitchA(config)# ip arp inspection vlan 10
SwitchA(config)# interface gigabitEthernet 1/1
SwitchA(config-if)# ip arp inspection trust

SwitchB(config)# ip arp inspection vlan 10
SwitchB(config)# interface gigabitEthernet 1/1
SwitchB(config-if)# ip arp inspection trust
```

Verification

```
SwitchA# show ip arp inspection interfaces
Interface              Trust State    Rate (pps)     Burst Interval
---------------        -----------    ----------     --------------
Gi1/1                  Trusted        None           N/A
Gi1/2                  Untrusted      15             1
Fa2/1                  Untrusted      15             1
Fa2/2                  Untrusted      15             1
```

```
SwitchA# show ip arp inspection vlan 10
Source Mac Validation          : Disabled
Destination Mac Validation     : Disabled
IP Address Validation          : Disabled
Vlan    Configuration  Operation ACL Match    Static ACL
----    -------------  --------- ---------    ----------
10      Enabled        Active
Vlan    ACL Logging    DHCP Logging
----    -----------    ------------
10      Deny           Deny
```

```
SwitchA# show ip dhcp snooping binding
MacAddress          IpAddress   Lease(sec)  Type           VLAN Interface
-----------------   ----------  ----------  -----------    ----  --------------
00:01:00:01:00:01   10.10.10.1  4995        dhcp-snooping  10   FastEthernet2/1
```

DAI Remarks

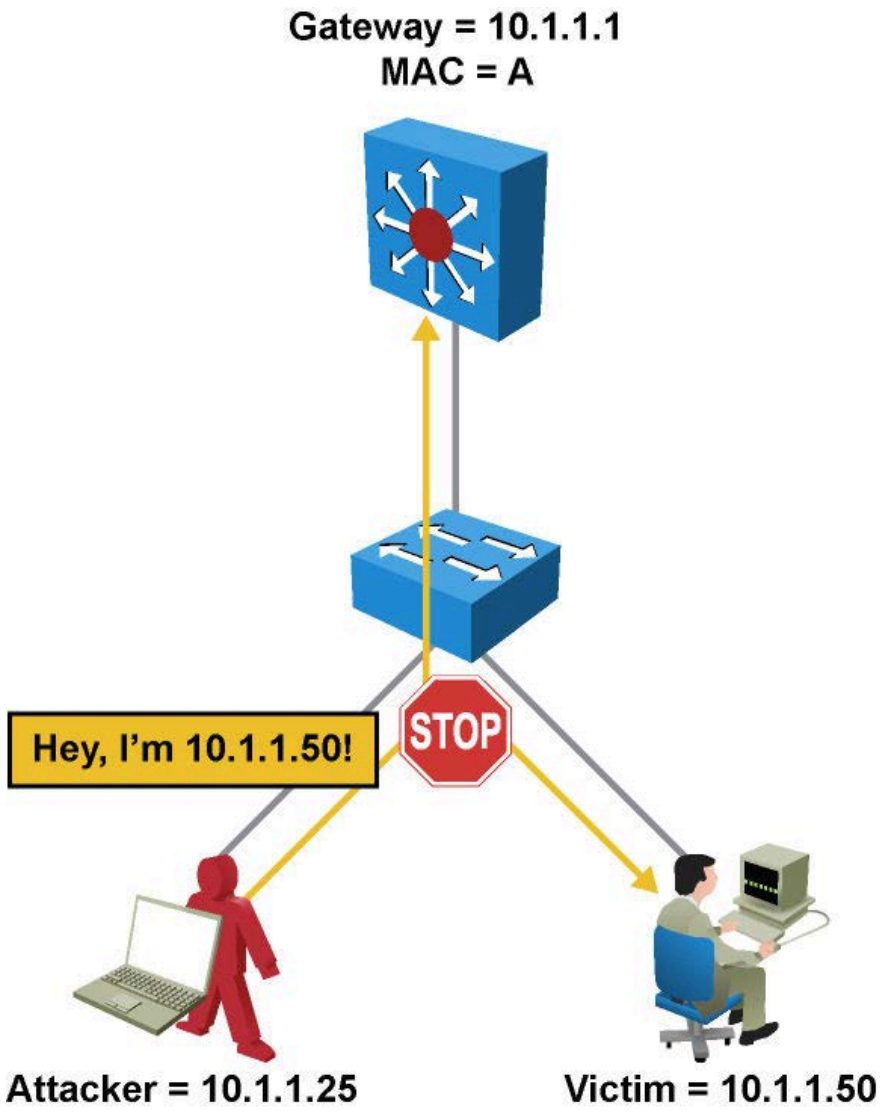
- DAI could be used without DHCP Snooping function using special ARP ACLs

```
Switch(config)# arp access-list ARP-V1
Switch(config-arp-acl)# permit ip host 1.1.1.1 mac host 0201.0203.0405
Switch(config-arp-acl)# exit
Switch(config)# ip arp inspection filter ARP-V1 vlan 1
```

IP Spoofing Attack
&
IP Source Guard

IP Spoofing and IP Source Guard

- Attacker impersonates a legitimate host on the network by spoofing the IP address of the victim
- **IP source guard (IPSG)** prevents a malicious host from attacking the network with a hijacked IP address with help of dynamically maintained per-port ACL's based on IP-to-MAC-to-switch port bindings
- IPSG works closely with DHCP snooping
  - At first, all IP traffic on the port is blocked except for DHCP packets captured by the DHCP snooping process
  - This restricts the client IP traffic to those source IP addresses configured in the binding; traffic with different source IP address is filtered out



Configuration

1) a) Enables IP Source Guard with source IP filtering

```
Switch(config-if)# ip verify source vlan dhcp-snooping
```

b) Enables IP Source Guard with source IP and source MAC address filtering

```
Switch(config-if)# ip verify source vlan dhcp-snooping port-
security
```

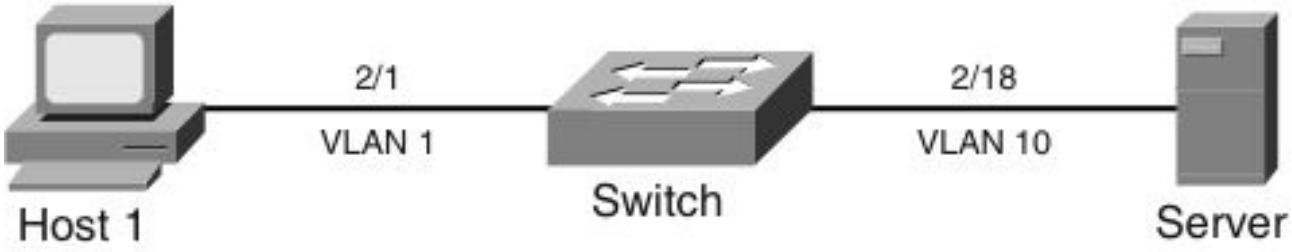
2) Optionally sets the rate limit for bad packets

```
Switch(config-if)# switchport portsecurity limit rate invalid N
```

3) Configures a static IP binding on the port

```
Switch(config)#
   ip source binding ipaddr ip vlan number interface IFACE
```

Example: IPSG




```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 1,10
Switch(config)# ip dhcp snooping verify mac-address
Switch(config)# ip source binding 0000.000a.000b vlan 10 10.1.10.11 int Fa2/18
Switch(config)# interface fastethernet 2/*
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# ip verify source vlan dhcp-snooping port-security
```

```
Switch# show ip source binding
MacAddress          IpAddress     Lease(sec)   Type                 VLAN   Interface
------------------ ------------ ---------- ------------  ----   ----------
00:02:B3:3F:3B:99  10.1.1.11     6522    dhcp-snooping      1      FastEthernet2/1
00:00:00:0A:00:0B  10.1.10.11    infinite    static                  10     FastEthernet2/18

Switch# show ip verify source
Interface Filter-type Filter-mode      IP-address          Mac-address          Vlan
--------- ----------- -----------      --------------- -----------------  ----------
Fa2/1       ip-mac      active         10.1.1.11           00:02:B3:3F:3B:99  1
Fa2/18      ip-mac      active         10.1.10.11          00:00:00:0a:00:0b  10
```

Securing Remote Access

Telnet Vulnerabilities

- With **Telnet**, all usernames, passwords, and data sent over the public network in clear text are vulnerable.
- A user with an account on the system could gain elevated privileges.
- A remote attacker could crash the Telnet service, preventing legitimate use of that service by performing a DoS attack such as opening too many bogus Telnet sessions.
- A remote attacker could find an enabled guest account that might be present anywhere within the trusted domains of the server.

Secure Shell

- **Secure Shell (SSH)** is a client/server protocol used to log in to another computer over a network to execute commands and to move files
- SSH provides strong authentication and secure communications over insecure channels
- It is a replacement for rlogin, rsh, rcp, and rdist in addition to Telnet
- Entire login session, including transmission of password, is encrypted; therefore, it is almost impossible for an outsider to collect passwords
- *Although SSH is secure, vendors' implementations of SSH might contain vulnerabilities!*
- SHS version 1 implementations are vulnerable to various security compromises; whenever possible, use SSH version 2 instead of SSH version 1

Configuring SSH

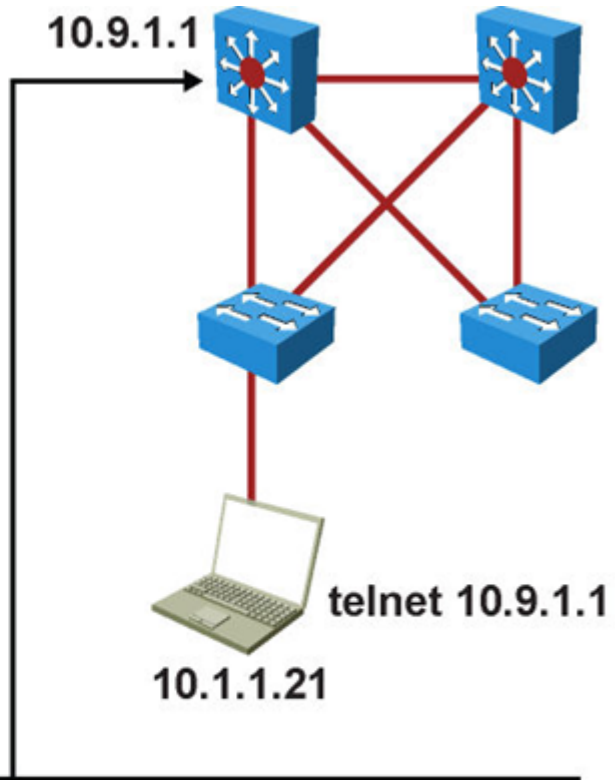
1) Configure a user with a password.
2) Configure the hostname and domain name.
3) Generate RSA keys.
4) Allow SSH transport on the vty lines.

- Configuration snippet

```
switch(config)# username xyz password abc123
switch(config)# ip domain-name xyz.com
switch(config)# crypto key generate rsa
switch(config)# ip ssh version 2
switch(config)# line vty 0 15
switch(config-line)# login local
switch(config-line)# transport input ssh
```

VTY Access Control Lists

- Cisco provides ACLs to permit or deny Telnet/SSH access to the vty ports of a switch
- **`access-class`** command…
  - …is used to filter incoming Telnet/SSH sessions by source address and to apply filtering to vty lines
  - **…**also applies standard IP ACL filtering to vty lines for outgoing Telnet/SSH sessions that originate from the switch
- You can apply vty ACLs to any combination of vty lines
  - The same ACL can be applied globally to all vty lines or separately to each vty line
- To configure vty ACLs on a Cisco switch, create a standard IP ACL and apply the ACL on the vty interfaces




```
sw(config)# access-list 100 permit ip 10.1.1.0 0.0.0.255 any
sw(config)# line vty 0 15
sw(config-line)# access-class 100 in
```

HTTP Secure Server

- A web interface is available to configure most switches
- The main weakness of the web interface is that it is not encrypted, and part of it does not offer any filtering
  - By default, it can be viewed by any user entering the switch IP address in a web browser address bar
- To protect the web service, you can take several steps:
  1) Configure username and password
  2) Configure domain name
  3) Generate RSA keys
  4) Enable HTTPS (SSL) server
  5) Configure HTTP authentication
  6) Configure an access list to limit access

Example: HTTPS Configuration

```
sw(config)# access-list 100 permit ip 10.1.9.0 0.0.0.255 any
sw(config)# username xyz password abc123
sw(config)# ip domain-name xyz.com
sw(config)# crypto key generate rsa
sw(config)# no ip http server
sw(config)# ip http secure-server
sw(config)# http access-class 100 in
sw(config)# http authentication local
```

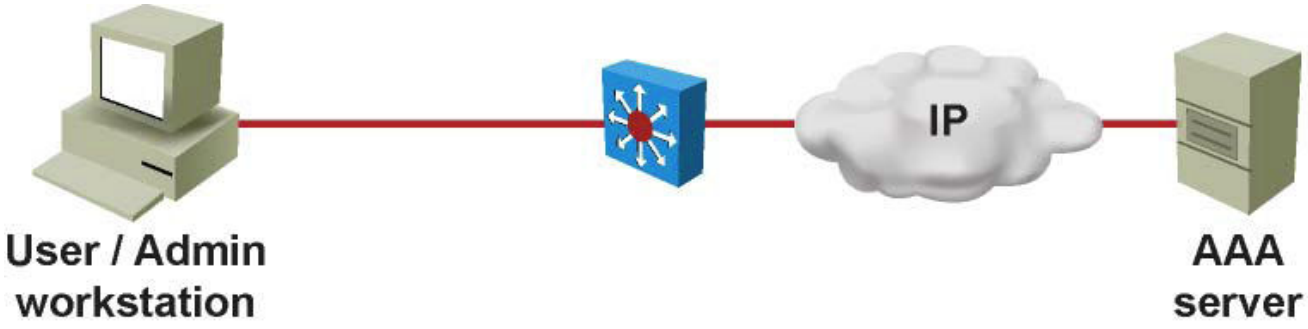
Authentication
Authorization
Accounting

AAA

- **Authentication**
  - User identification
  - Login and password dialog
  - Challenge and response
- **Authorization**
  - One-time authorization or
  - Authorization for each service on a per-user account list or a user group basis
- **Accounting**
  - Start and stop times
  - Executed commands
  - Number of packets
  - Number of bytes
- **Authentication, Authorization, Accounting (AAA)** is an architectural framework for configuring a set of three independent security functions in a consistent manner

AAA Process

- The AAA authorization process works by contacting a common, centralized database of a set of attributes that describe the network user's authorized services
- The centralized server returns a result of allowed services in question to execute the user's actual capabilities and restrictions
- The remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating their **attribute-value pairs (AVP)**



RADIUS vs. TACACS AVPs

Attribute	Type of Value
User-Name	String
Password	String
CHAP-Password	String
Client-Id	IP address
Login-Host	IP address
Login-Service	Integer
Login-TCP-Port	Integer

Attribute	Type of Value
Inacl	Integer
Addr-pool	String
Addr	IP address
Idletime	Integer
Protocol	Keyword
Timeout	Integer
Outacl	Integer

AAA Models

- Cisco devices supports two AAA models
- **Older one**
  - Authentication and authorization only against local database
  - Minimal accounting support
- **Newer one**
  - Complex separate configuration for each AAA part

- *Beware of AAA misconfiguration, because you migth end up with locked devices with no access options!*

New AAA model

- New AAA model has following assumptions
  - On one side we have certain set of services that could leverage specific mechanism in order to control access
    - E.g. dot1x, enable, login, ppp
  - On the other side we have different databases with user account and their privilieges
    - E.g. RADIUS, TACACS, local database
  - The goal is to specify which database to use when attempting to perform an action in frame of certain service
- E.g.
  - Console access would be protected by local database
  - SSH remote login would be protected by TACACS server with IP 1.1.1.1
  - Access to Ethernet switchport would be granted upon RADIUS authentication against server 2.2.2.2

Configuring Authentication

- Activate new AAA model:

```
Router(config)# aaa new-model
```

- Create a new list of authentication methods:

```
Router(config)#
  aaa authentication {ppp|dot1x|enable|login} list-name
  method1 [method2…]
```

- Configuration snippet depends on use-case:

```
Router(config)# aaa new-model
Router(config)# aaa authentication login L_LOCAL local
Router(config)# line vty 0 15
Router(config-line)# login authentication L_LOCAL
```

Configuring Authorization

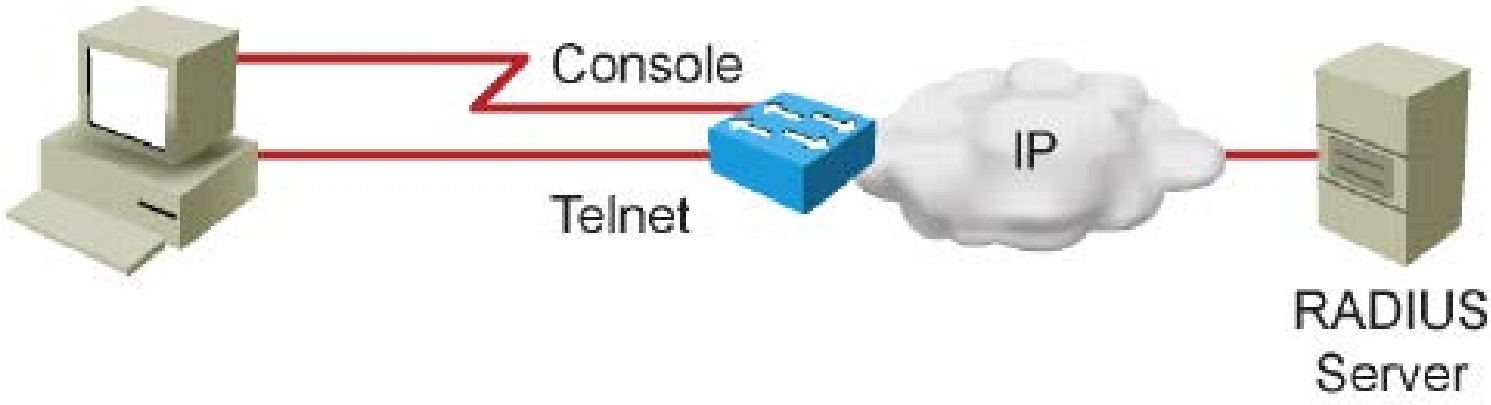
- With new AAA model, authorization is strictly separated from authentication
  - *Beware of authenticated user without any privileges!*
- Create a new list of authorization methods:

```
Router(config)#
  aaa authorization {auth-proxy | network | exec | commands
  level | reverse-access | configuration | ipmobile} {default
  | list-name} [method1 [method2...]] authorization {arap |
  commands level | exec | reverse-access} {default | list-
  name}
```

- Configuration snippet depends on use-case:

```
Router(config)# aaa new-model
Router(config)# aaa authorization exec E_LOCAL local
Router(config)# line vty 0 15
Router(config-line)# authorization exec E_LOCAL
```

Example: Authentication + Authorization




```
(conf)# aaa new-model
(conf)# username admin password cisco privilege 15
(conf)# aaa authentication login L_RAD+L group radius local line
(conf)# aaa authentication login L_NONE none
(conf)# aaa authorization exec E_RAD+L group radius local
(conf)# radius-server host 1.2.3.4 auth-port 1812 acct-port 1813 key PAS
(conf)# line vty 0 15
(conf-line)# login authentication L_RAD+L
(conf-line)# authorization exec E_RAD+L
(conf-line)# password class
(conf)# line console 0
(conf-line)# login authentication L_NONE
```

Configuring Accounting

- Create a new list of accounting methods:

```
Router(config)#
  aaa accounting {system | network | exec | connection |
  commands level} {default | list-name} {start-stop | stop-
  only | none} [method1 [method2...]]
```

- Apply the accounting method to an interface or lines using the command:

```
Router(config)#
  accounting {arap | commands level | connection | exec}
  {default | list-name}
```

Accounting Types

- **Network accounting:** Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts
- **Connection accounting:** Provides information about all outbound connections made from the network, such as Telnet and rlogin
- **EXEC accounting:** Provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number from which the call originated
- **System accounting:** Provides information about all system-level events (for example, when the system reboots and when accounting is turned on or off)
- **Command accounting:** Provides information about the EXEC shell commands for a specified privilege level executed on a network access server
- **Resource accounting:** Provides start and stop record support for calls that have passed user authentication

Example: Accounting

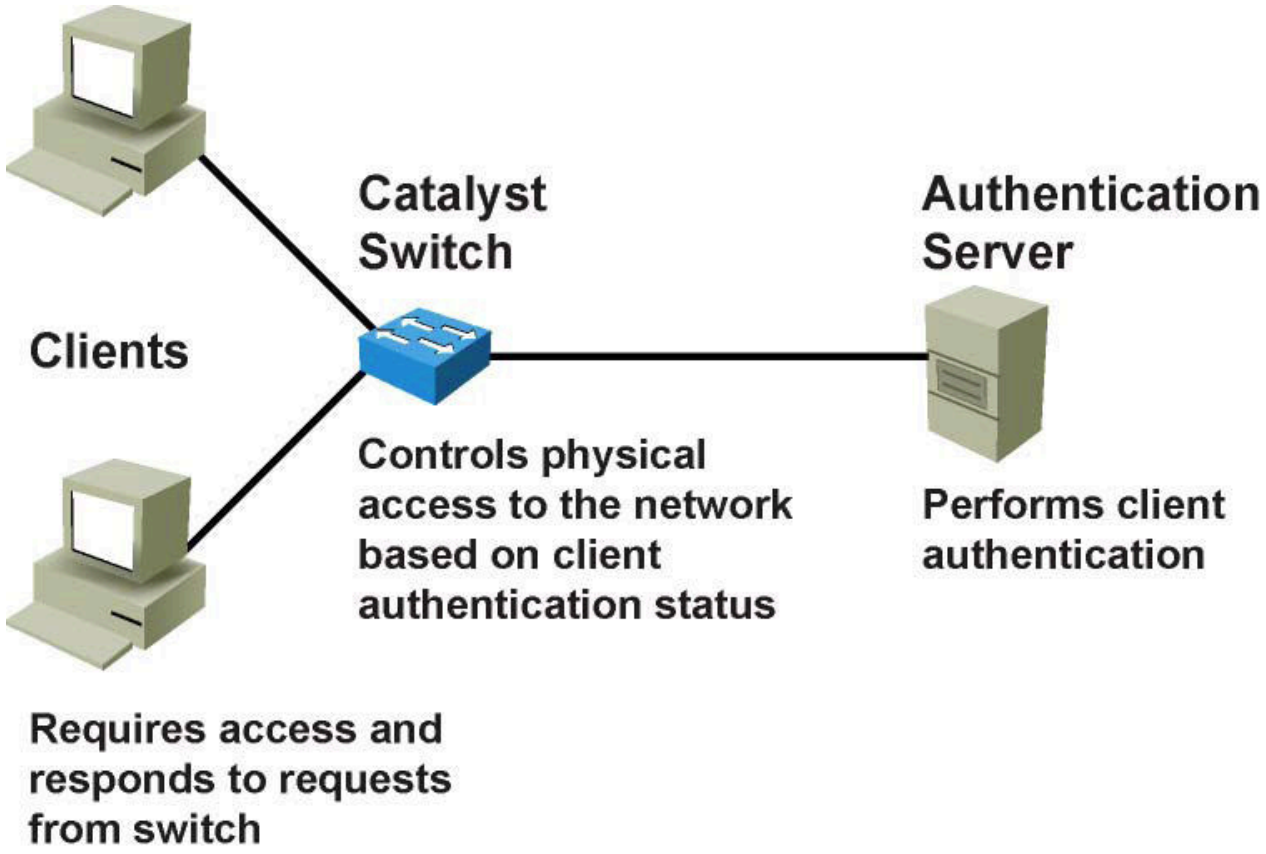
- This configuration example illustrates configuring AAA authorization for users via VTY access for shell commands.
- To allow users to access the functions they request as long as they have been authenticated, use **if-authenticated** method keyword

```
Switch(config)# aaa new-model
Switch(config)# aaa authentication login default tacacs+
Switch(config)# tacacs-server host 192.168.100.100
Switch(config)# aaa authorization commands 0 default if-
authenticated group tacacs+
Switch(config)# aaa accounting exec default start-stop group tacacs+
Switch(config)# line vty 0 4
Switch(config-line)# authorization commands 0 default
Switch(config-line)# accounting exec default
```

802.1X

802.1X

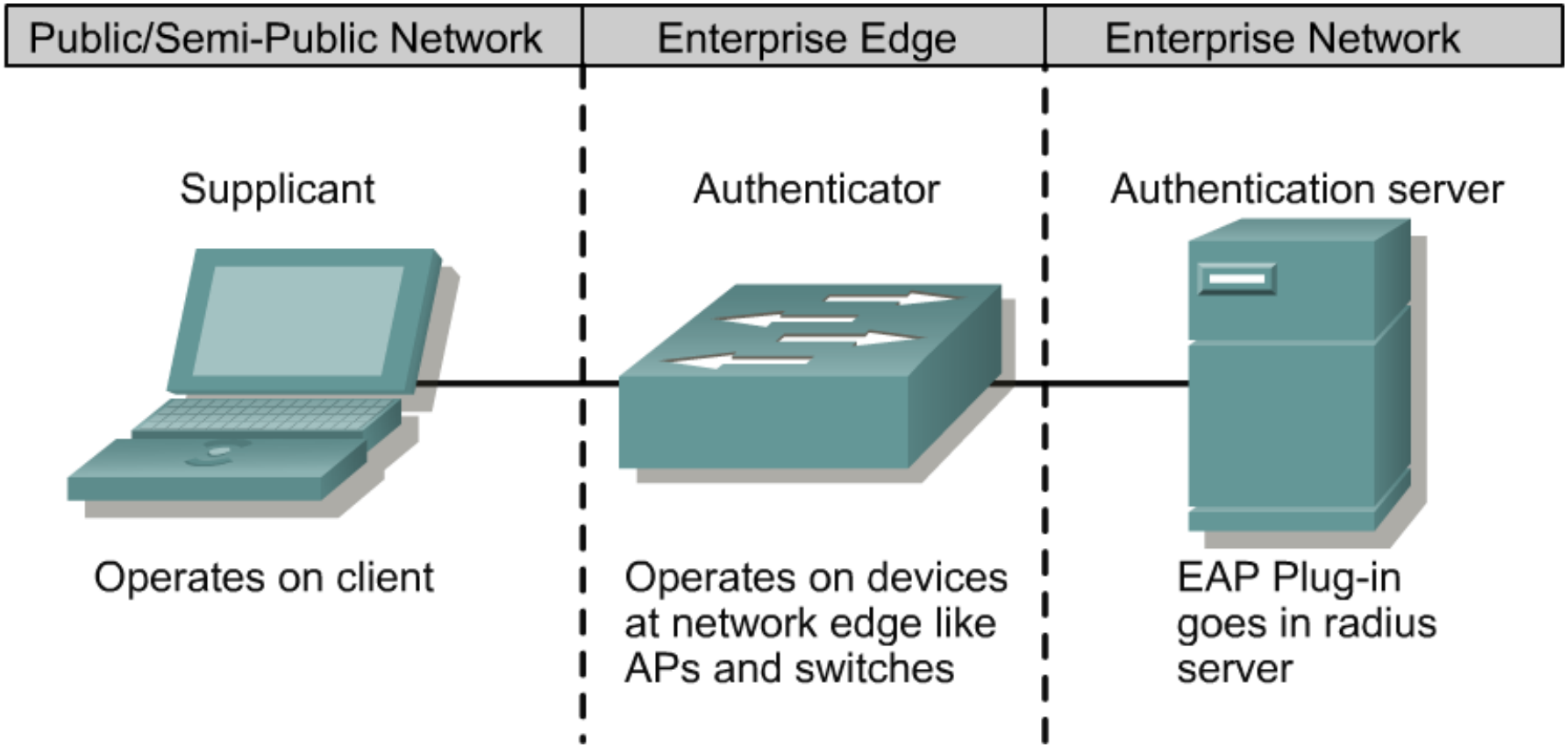
- 802.1X authetication may ask for login information
  - `IF` user successfuly authenticates `THEN` switch unblocks port `ELSE` port remain in quarantine or guest VLAN



Terms

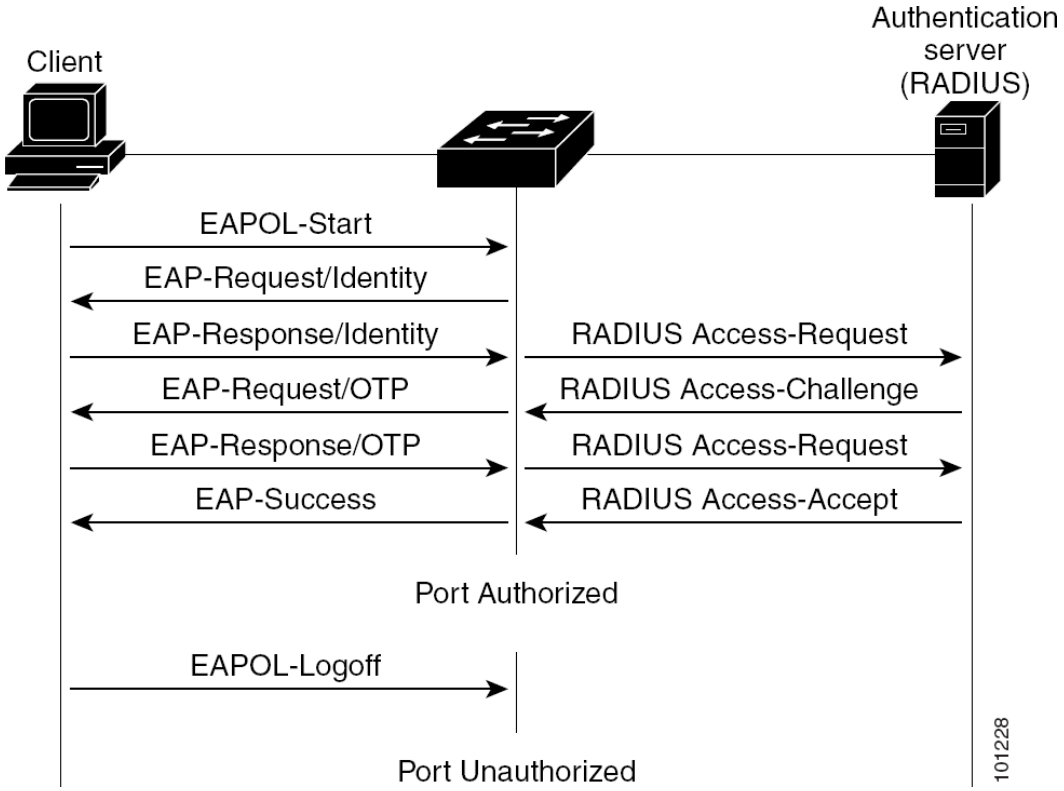
- **Supplicant**
  - SW client on PC that is responsible for sending authentication data in proper form
- **Authenticator**
  - Device to which PC is trying to connect and which requires authentication of PC
- **Authentication Server**
  - Contains user database with their privilleges
- **Extensible Authentication Protocol (EAP)**
  - Generic protocol for passing authentication data in unified form, specified by RFC 3748
  - Only EAP traffic is allowed on unauthenticated ports
- **RADIUS**
  - Authentication protocol between network access serves (NAS) and authentication server, specified by RFC 2865
  - Colaborates with EAP (RFC 3579)
- **802.1X**
  - IEEE standard for port-base authentication using EAP messages inside Ethernet frames (EAP over LAN = EAPOL) and RADIUS

Terms Illustration



Process

1) Client sends EAPOL-START
2) Switch asks for initial login information from client
3) EAP answer is encapsulated into RADIUS and send towards authentication server by switch
4) RADIUS may immediately authenticate user or more complicated request-response message exchange follows
5) After successful implementation RADIUS sends Access-Accept and subsequently switch grants access to client with EAP-Success



Configuration

1) Enable AAA

```
Switch(config)# aaa new-model
```

2) Create an 802.1X port-based authentication method list

```
Switch(config)#
   aaa authentication dot1x {default} method1 [method2...]
```

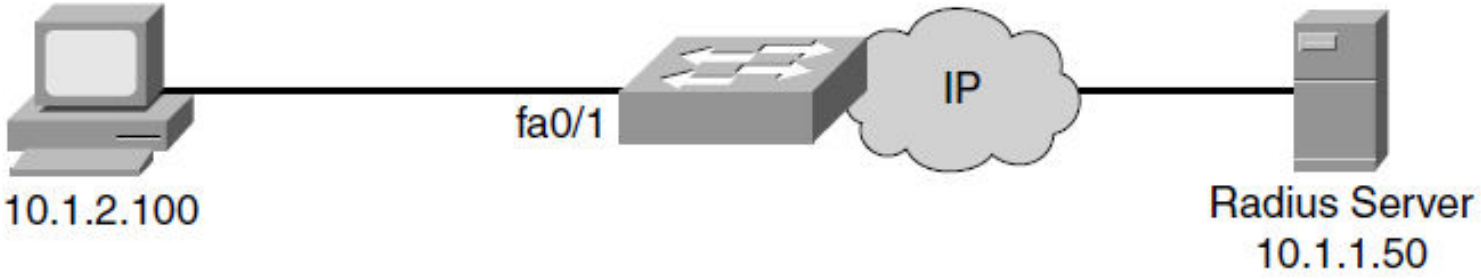
3) Globally enable 802.1X port-based authentication

```
Switch(config)# dot1x system-auth-control
```

4) Enable 802.1X port-based authentication on the interface

```
Switch(config-if)# dot1x port-control { auto | force-
authorized | force-unauthorized }
```

Example: 802.1X




```
aaa new-model
aaa authentication dot1x default group radius

! The next line is unnecessary when we do not want to use dynamic VLANs
aaa authorization network default group radius

radius-server host 1.2.3.4 auth-port 1812 acct-port 1813 key HESLO
dot1x system-auth-control
interface FastEthernet 0/1
 switchport mode access
 dot1x port-control auto
 ! Newer switches uses following two commands instead of previous one
 ! authentication port-control auto
 ! dot1x pae authenticator
```

Remarks

- 802.1X allows multiple dynamically assigned VLANs
  - Guest VLAN for clients without 802.1X support
  - Restricted VLAN for clients supporting 802.1X which failed to authenticate (i.e. Auth-Fail VLAN)
- 802.1X might cause troubles whenever there are more devices connected to target port
  - 2950s and 3550s CANNOT authenticate individual hosts, thus port is either always opened or closed for anyone
  - 2960s and 3560s supports individual authentication since IOS 12.2(50)SE (i.e. multiauth mode)
- RADIUS server configuration is crucial for 802.1X
  - More advanced means of authentication (e.g. PEAP, EAP-TLS) require X.509 certificates for servers or even clients

Where to Go Next?

- Catalyst 3560 Command Reference

  www.cisco.com/en/US/partner/docs/switches/lan/catalyst3560/software/release/12.2_55_se/command/reference/3560_cr.html

- Configuring Port Security:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/swtrafc.html#wp1038501

- Configuring IEEE 802.1X:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/sw8021x.html

- Configuring DAI:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/swdynarp.html

- Configuring IP Source Guard:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/swdhcp82.html

- Configuring EEM:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/sweem.html

Attack's infographic comes from Marek Lomnický,
slides adapted by Vladimír Veselý,
partially from official course materials
but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

The last update: 2013-10-10