University of Connecticut
Computer Science and Engineering
CSE 4402/5095: Network Security

Layer-2 ('LAN') Security

©Amir Herzberg

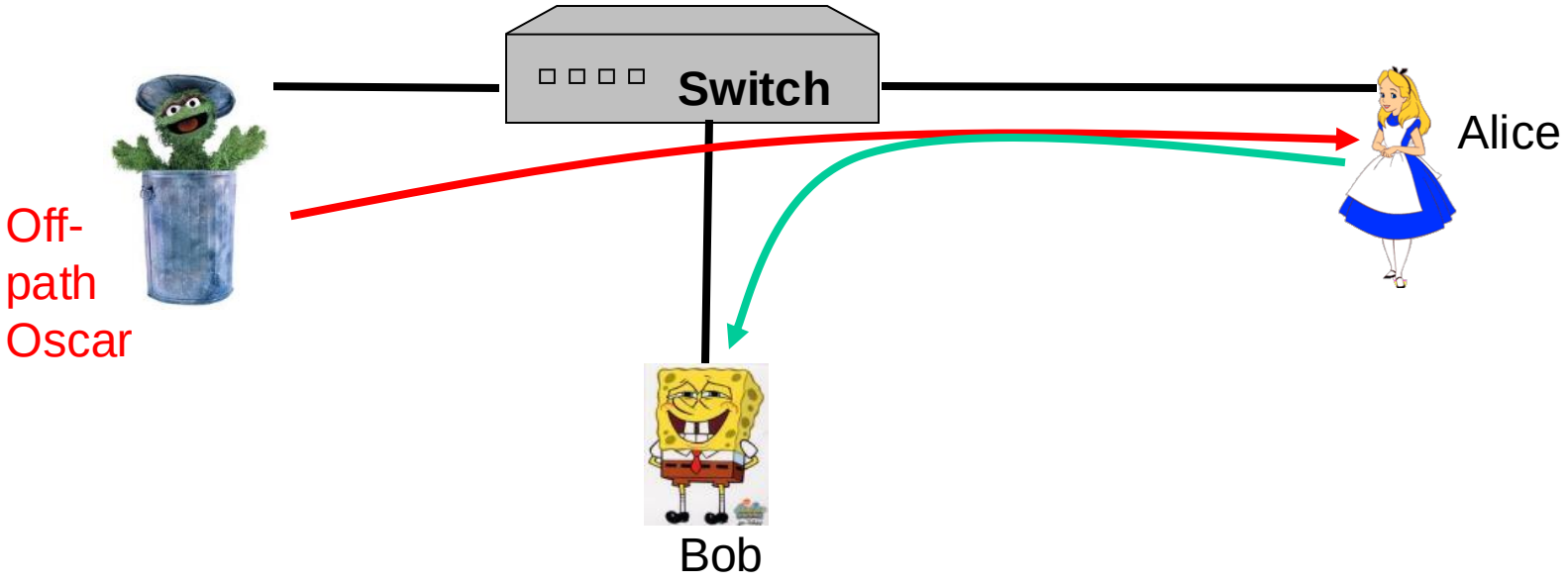
Version of Spring '23

Sniffing: Eavesdropping on Shared Media

- No special hardware necessary: 'Promiscuous mode'
  - Listen to packets for all destinations
  - Available with many network adapters
  - Long-range sniffing – with special (low-cost) hardware
- Easy – with (unencrypted) access to shared media
- Rare: shared wired media
- Common:
  - Wireless (WiFi, etc.)
    - Many (most?) wireless networking new use some cryptography
      - Often: vulnerable (e.g., WEP, WPA-1, WPA-2)
    - Sniffing may provide ciphertext, allow cryptanalysis attacks
  - Switched Ethernet
    - Traffic isolation … ?

Switches and Traffic Isolation

- Packets are broadcasted inside segments
  - Often, segments are wireless (or just contain one wired host)
- Traffic isolation: forward only as needed
  - By learning the link addresses in each segment
  - Goals: performance and security
- MITM on specific segment, off-path on others



Identifiers in Switched Networks

Network Layer: IP address (1.2.3.4 / 6.6.6.6)

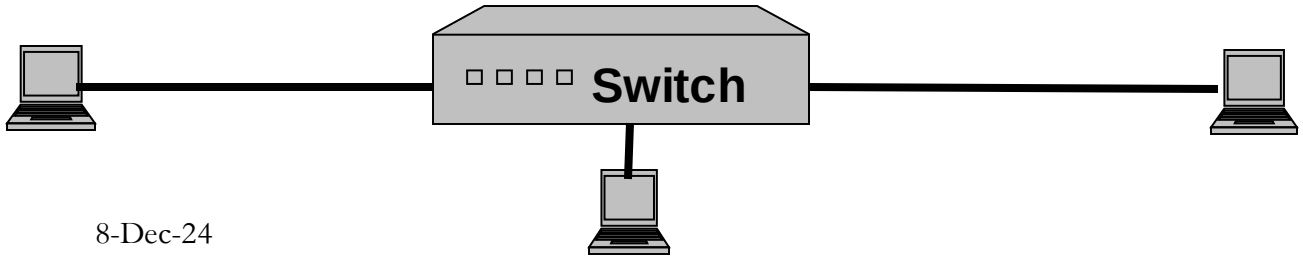
- Provided by DHCP: IP of host, gateway and resolver
- Network-part of IP used to route to dest network
- Resolved to MAC address by ARP

MAC (or LAN or physical or Ethernet) address:

- To identify source & destination on same network
- Most LANs: 48 bits, global address space
  - Special broadcast address – send to all nodes
- Mapped to interface – if known (from learning / spanning tree)

Interface identifier

- Identifies interface of switch – to forward to correct host

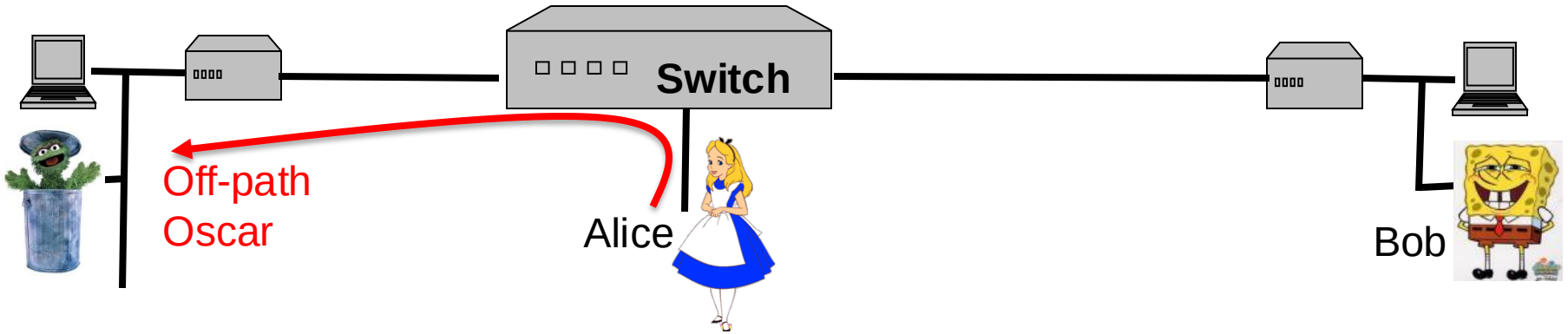


Steps of LAN communication

- DHCP: host connects
  - Receives IP for host, also for GW, DNS resolver
- ARP: find (resolve) MAC address from IP
  - Including of resolver (to find hosts), GW
  - Each host maintains its own ARP table
- Learning: find interface for each MAC address
  - Until known: send packets to all interfaces
  - When known: keep mapping in **switch table**
    - Hosts have only one interface
- DNS: find IP of destination (from name)
  - Discussed later (not specific to LAN)

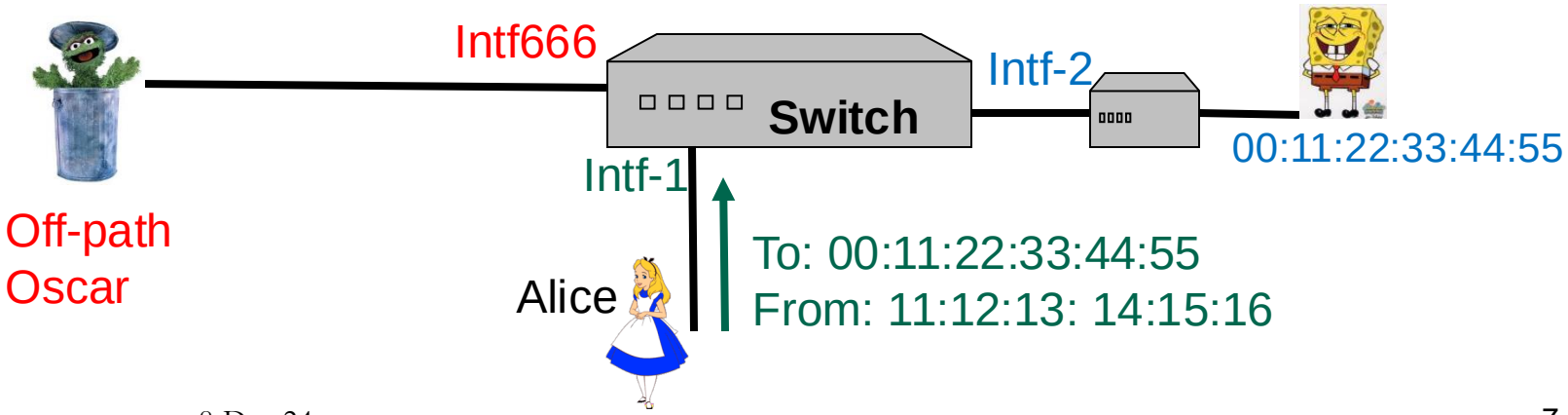
From Off-path to MitM

- Poisoning attacks: map traffic to Oscar
  - Interface poisoning: 00:11:22:33:44:55 → Intf666
  - DHCP poisoning: gateway/resolver → 6.6.6.6
  - ARP poisoning: 1.2.3.4 → 66:66:66:66:66:66
  - (Later: DNS poisoning: bob.com → 6.6.6.6)
- Or, degradation attacks: some switches broadcast if MAC table is too large
  - Use of DoS to foil defenses



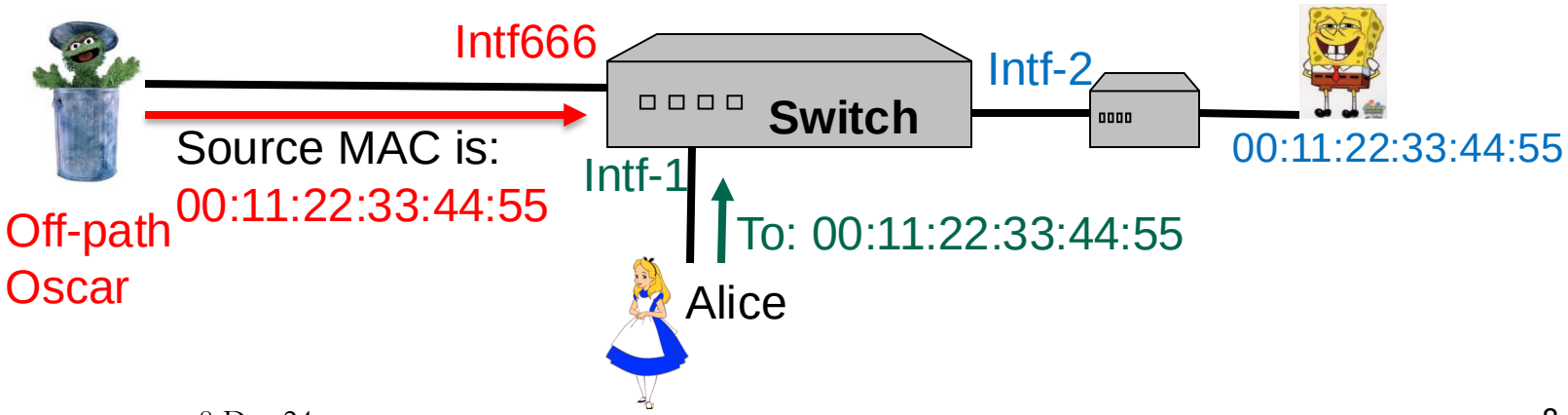
Switch Learning Mechanism

- Switch has multiple interfaces
- Receives frame (`packet') from one interface
  - Update table: switch(source-MAC)←interface
  - This is called `learning'
- Then, forward – to which interface(s)?
  - To interface from table: switch(destination-MAC)
  - No entry? Forward to all interfaces (broadcast)



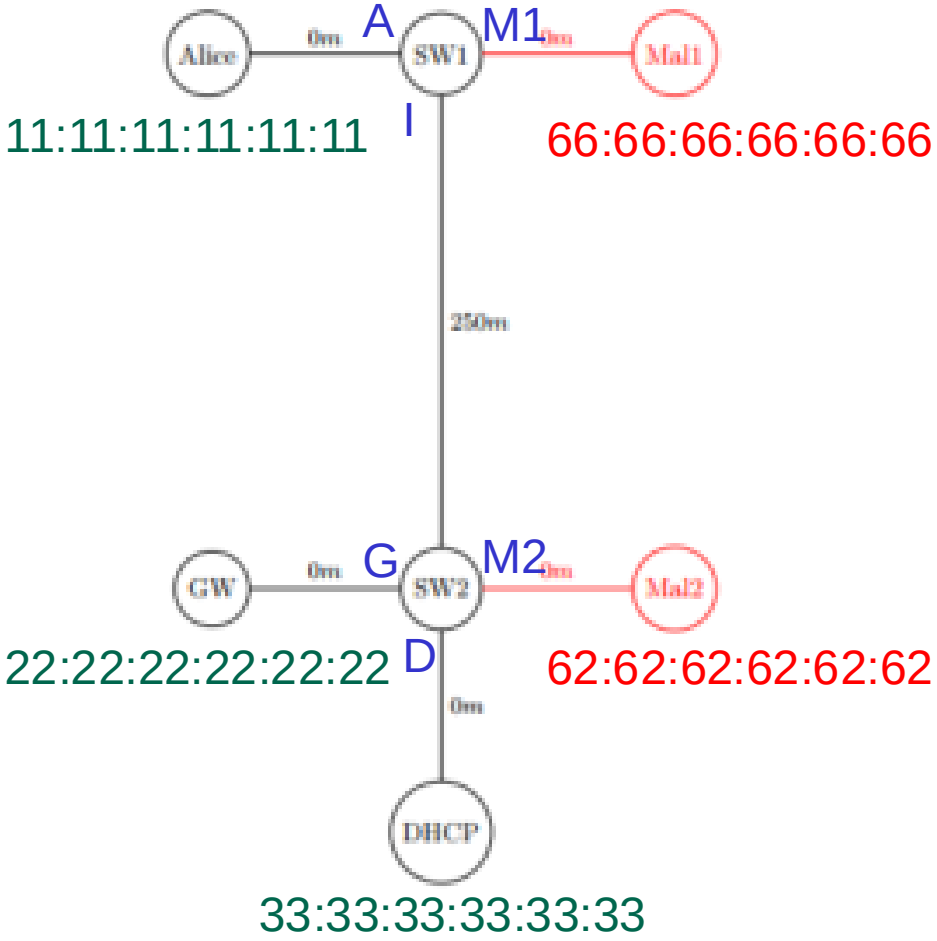
Interface Poisoning Attack

- Interface poisoning: 00:11:22:33:44:55 → Intf666
- Simple: just send packet with spoofed MAC!
- But reset upon 1st packet from honest source ☹
  - Not always- depends on network configuration: see exercise

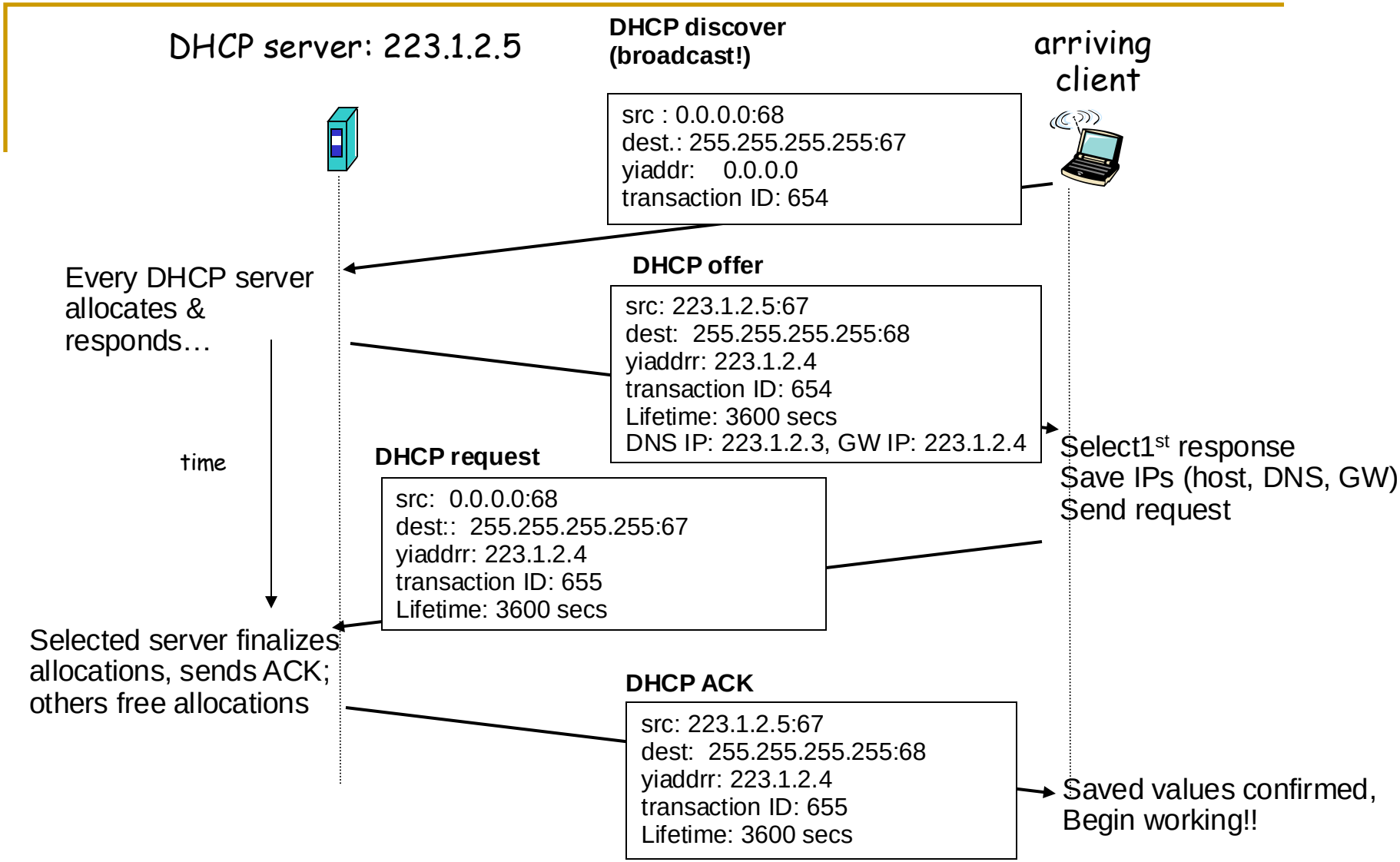


Exercise: Interface Poisoning MitM

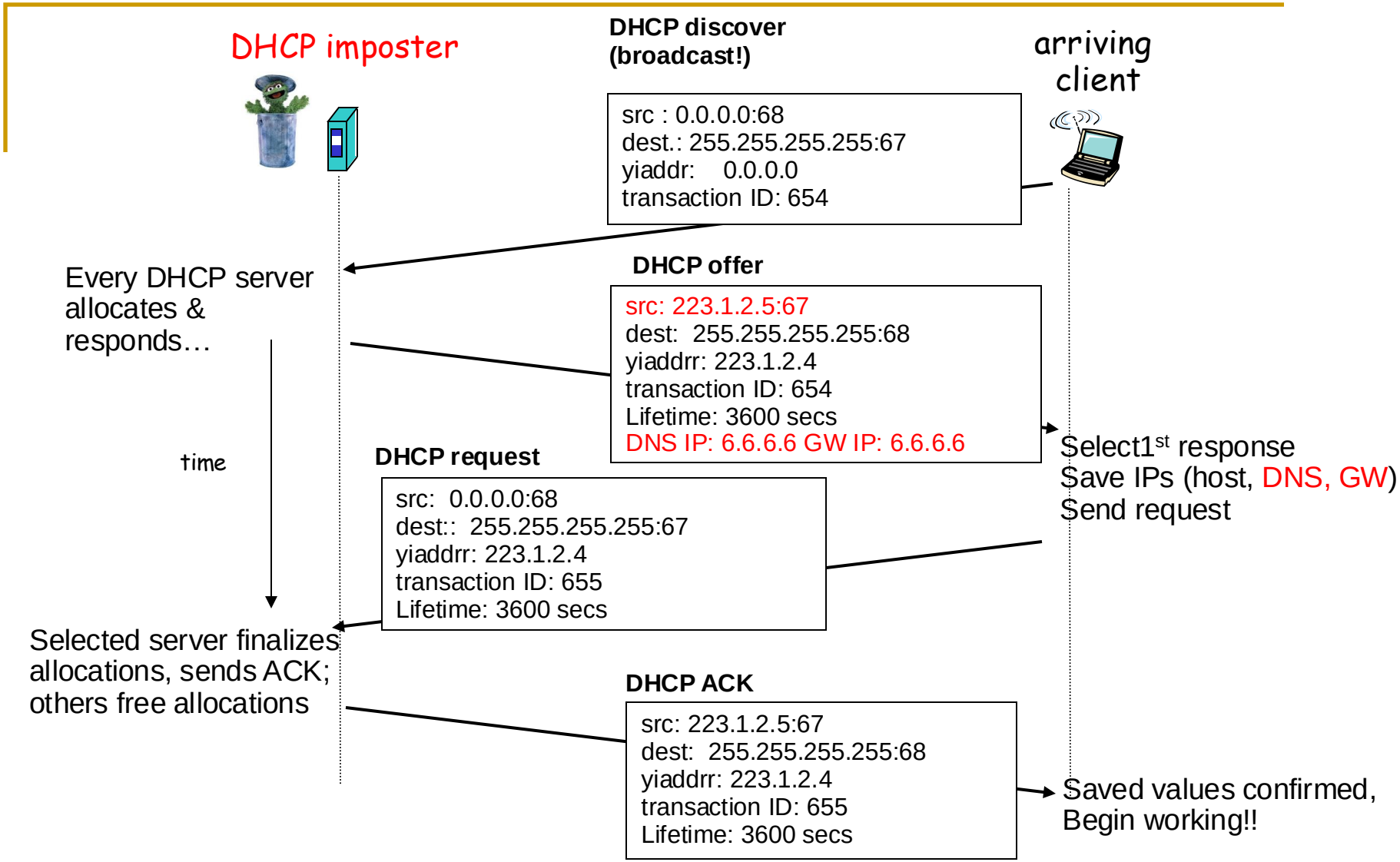
- Demonstrate MitM between Alice and GW (Internet), using interface poisoning
  - Hint: may use both Mal1 and Mal2



DHCP : Dynamic Host Configuration Protocol

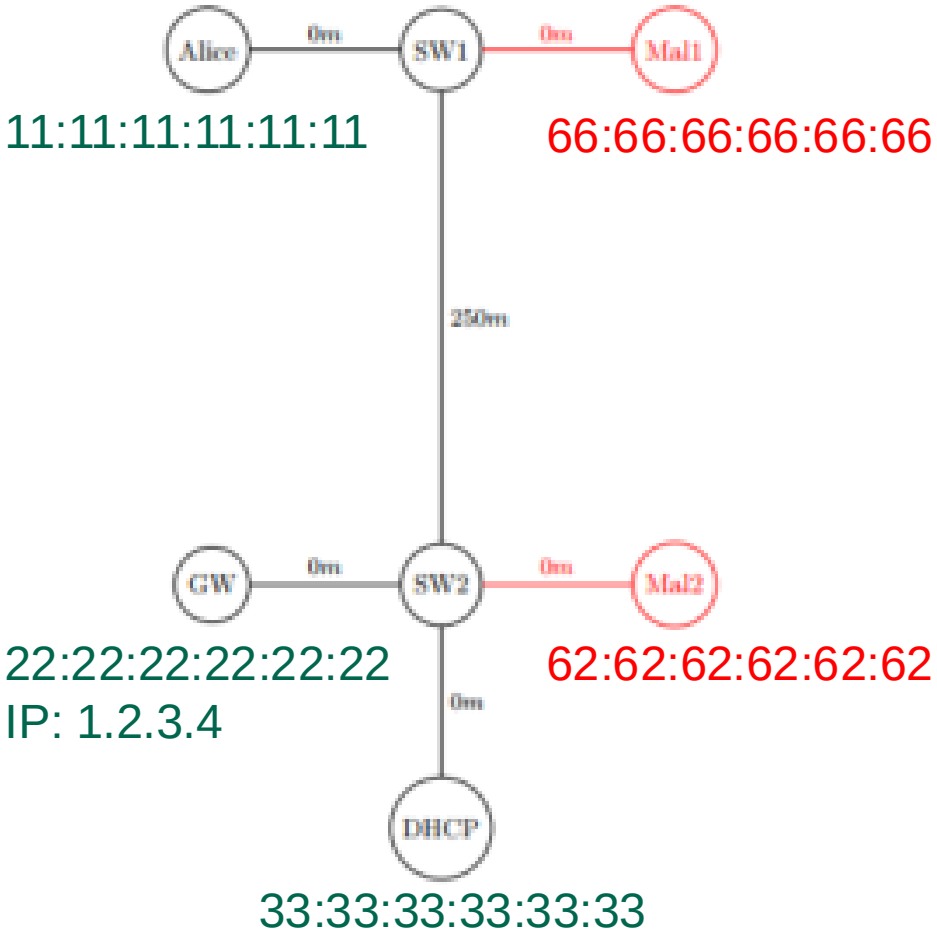


DHCP Poisoning



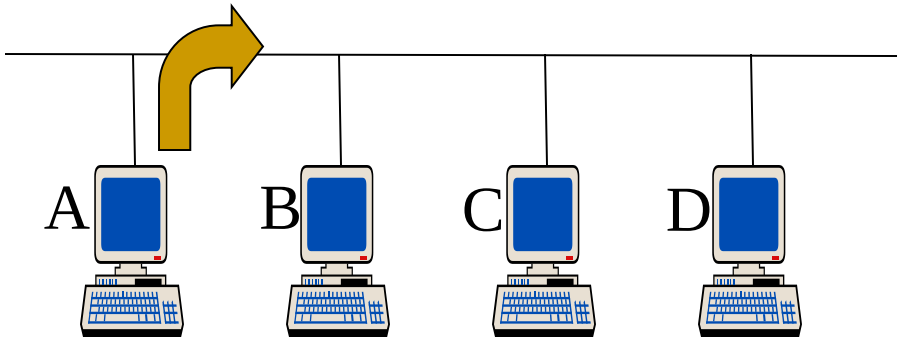
Exercise: DHCP Poisoning MitM

- Demonstrate MitM between Alice and GW (Internet), using DHCP poisoning
  - Hint: assume Alice just connected



Address Resolution Protocol (ARP)

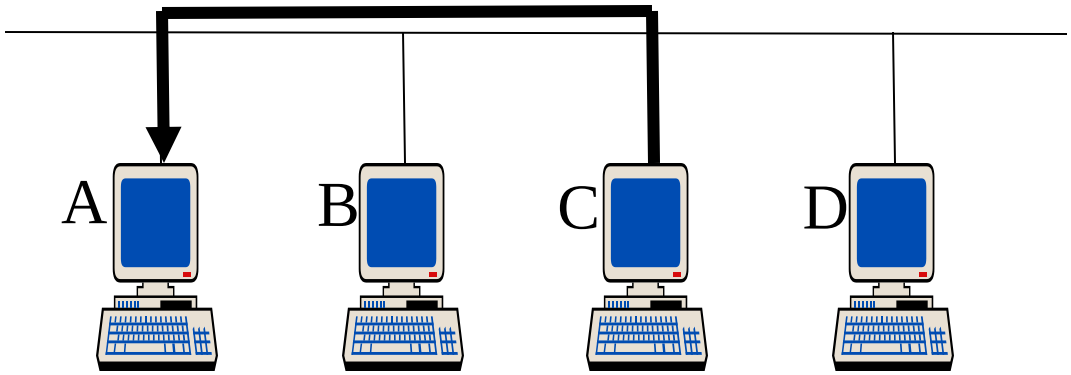
Broadcast Request: Sender IP, Sender MAC, Target IP



C learns A's IP, MAC
B, D could also learn, but usually don't (since they may not send to A). [some OS do]

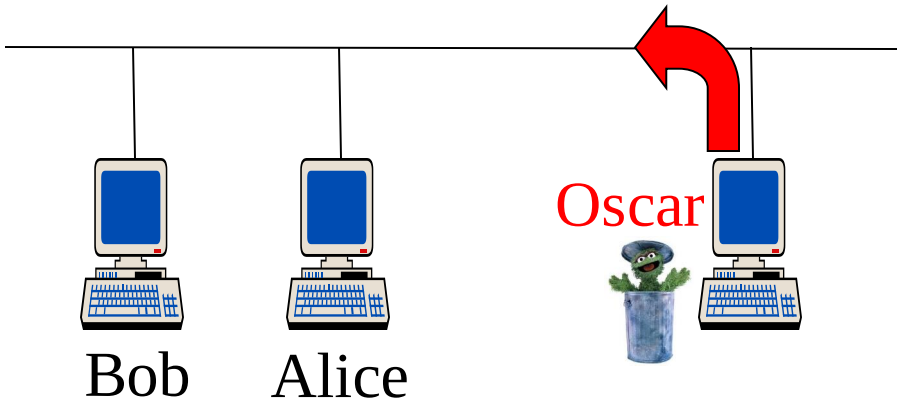
Unicast Response

A learns C's IP, MAC



ARP Poisoning: by Spoofed Request

ARP Request: from: (Bob's IP, Oscar's MAC), to: Alice's IP



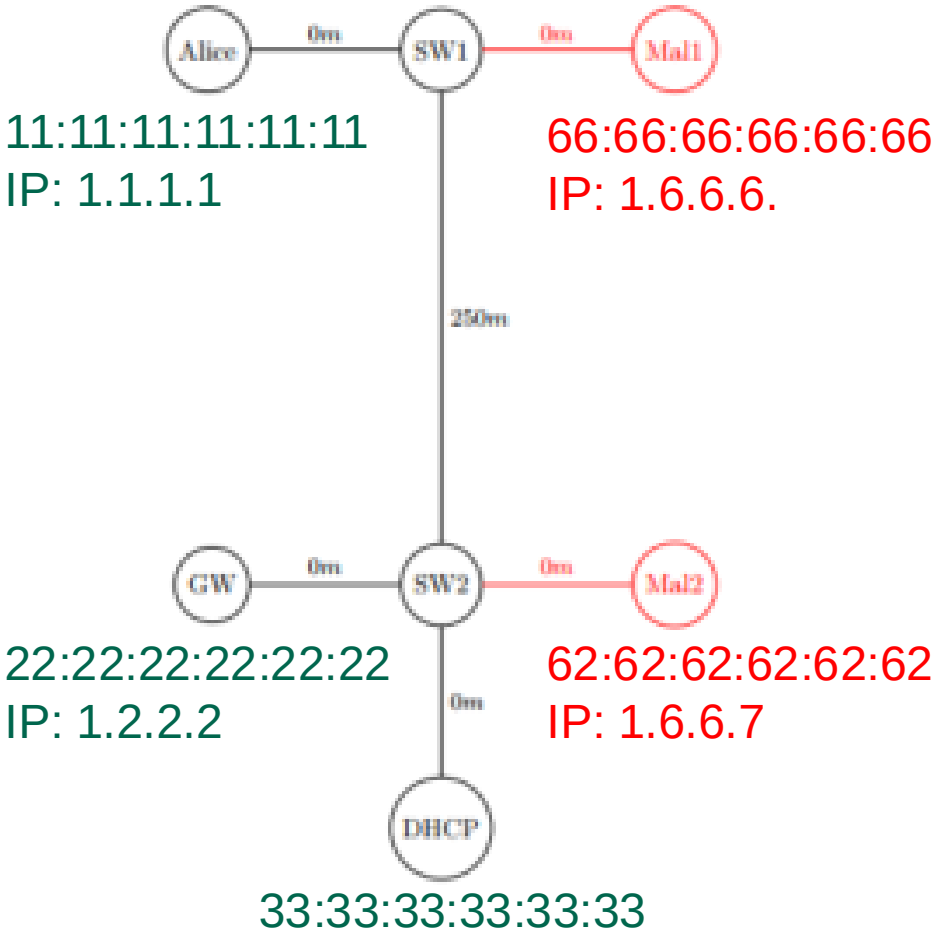
Many hosts (Alice) map Bob's IP to Oscar's MAC (for efficiency)

Bob often ignores (`not me')

➔ Hosts should not add to ARP table mapping from ARP requests

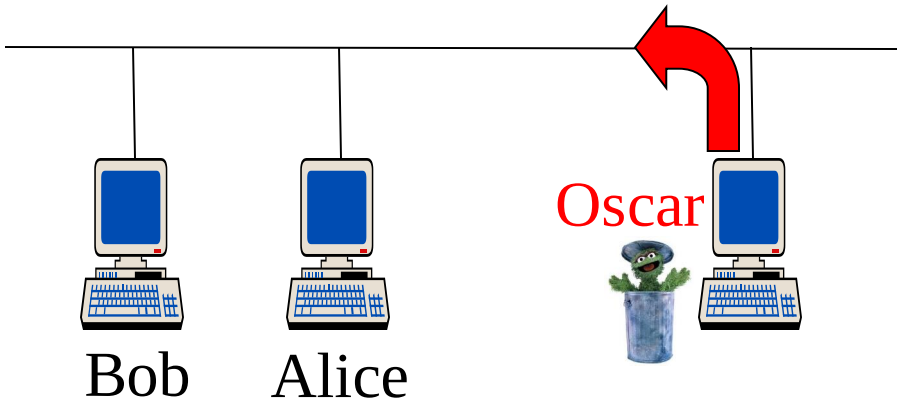
Exercise: ARP Poisoning MitM (1)

- Demonstrate MitM between Alice and GW (Internet), using spoofed ARP request



ARP Poisoning: by Gratuitous Response

ARP Response: from: (Bob's IP, Oscar's MAC), to: Alice's IP

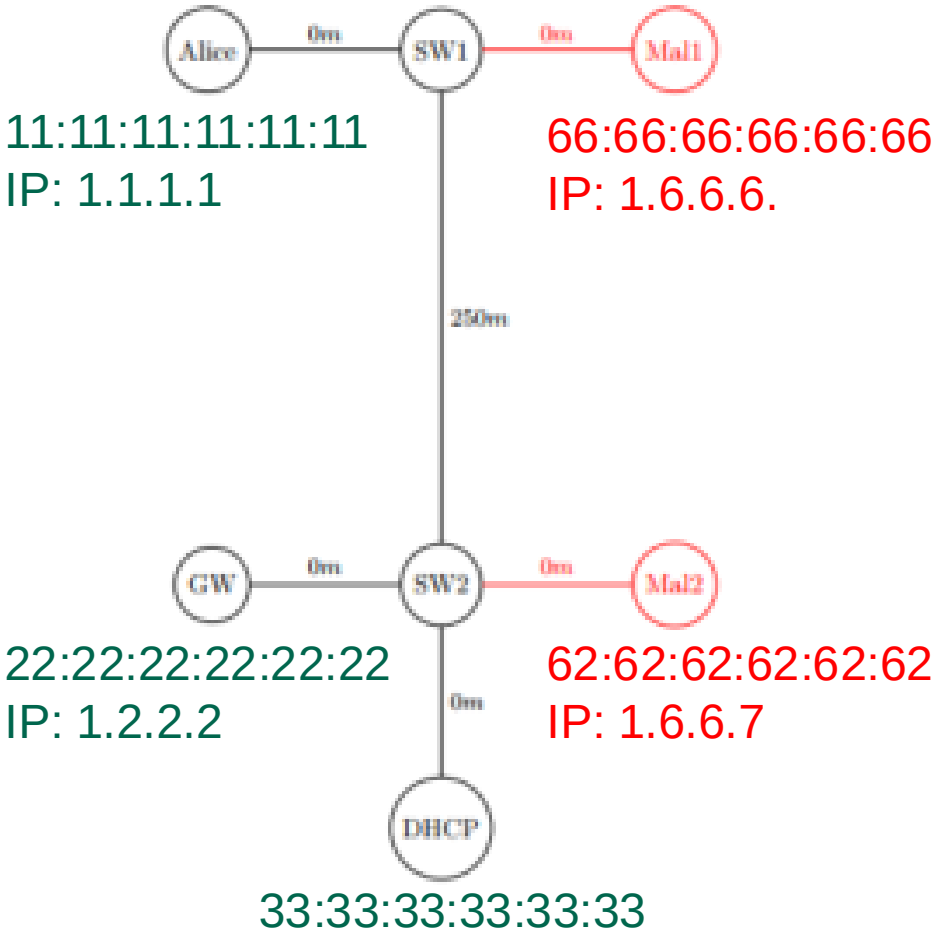


Some hosts do not keep state in ARP, and handle gratuitous response just like solicited response!!

➔ Hosts should ignore/alert on gratuitous ARP response

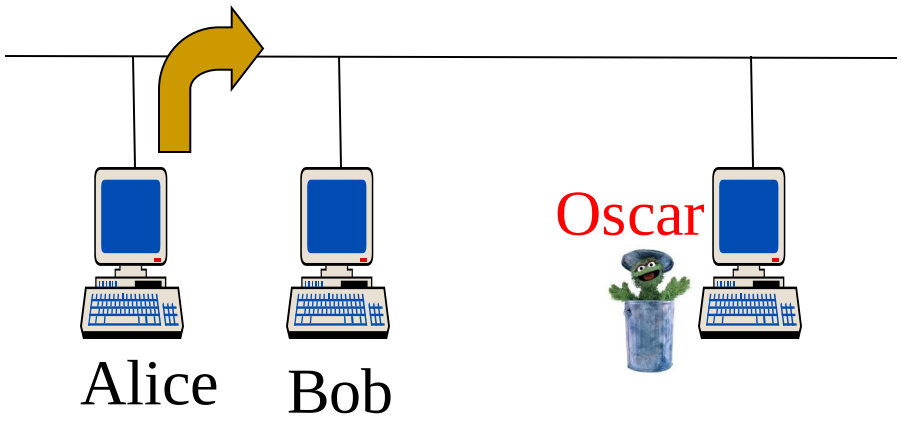
Exercise: ARP Poisoning MitM (2)

- Demonstrate MitM between Alice and GW (Internet), using gratuitous ARP response



ARP Poisoning by Responding

Broadcast Request: from: (Alice's IP, MAC), to: Bob's IP



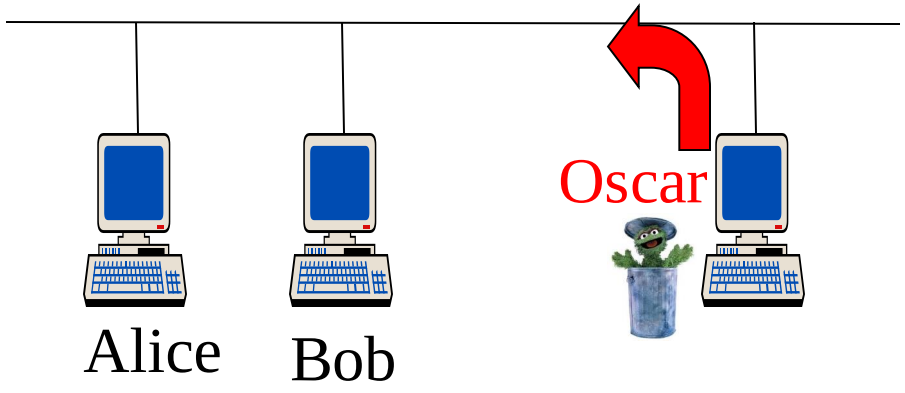
Oscar sniffs request, sends spoofed response

To win over Bob: abuse MAC's collision avoidance

Or send response before req!

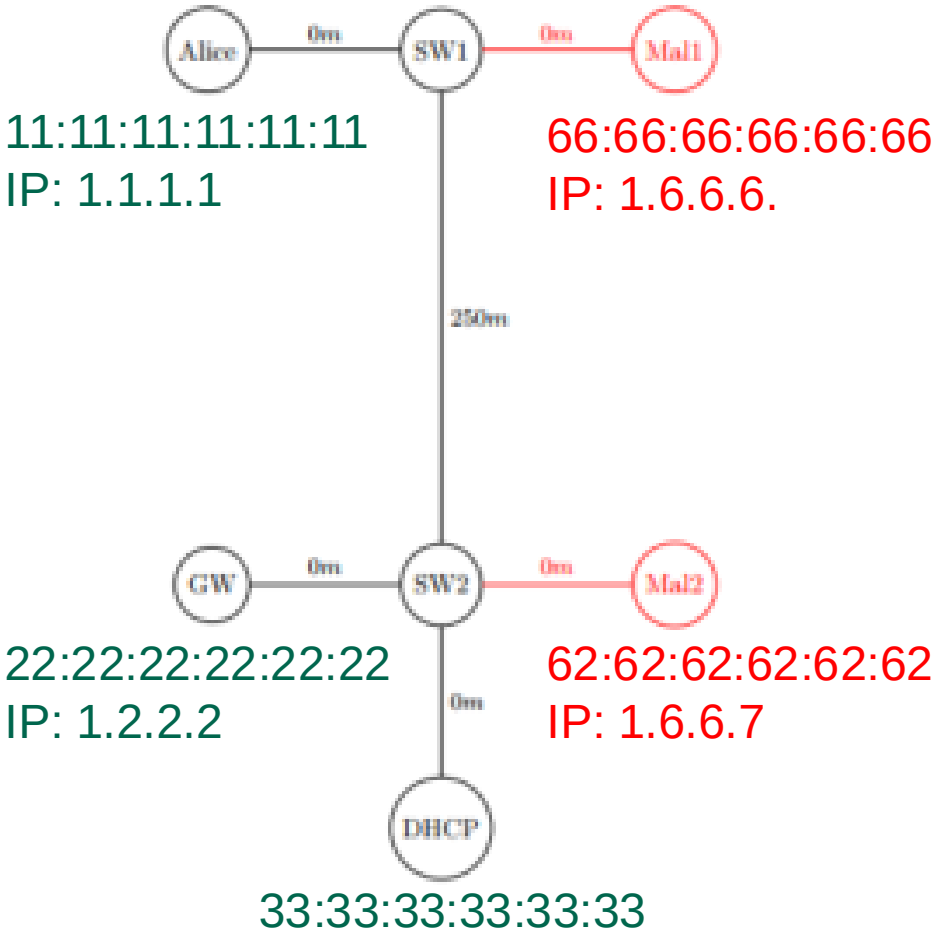
Unicast Spoofed Response

Source: Bob
MAC: Oscar's
Dest: Alice



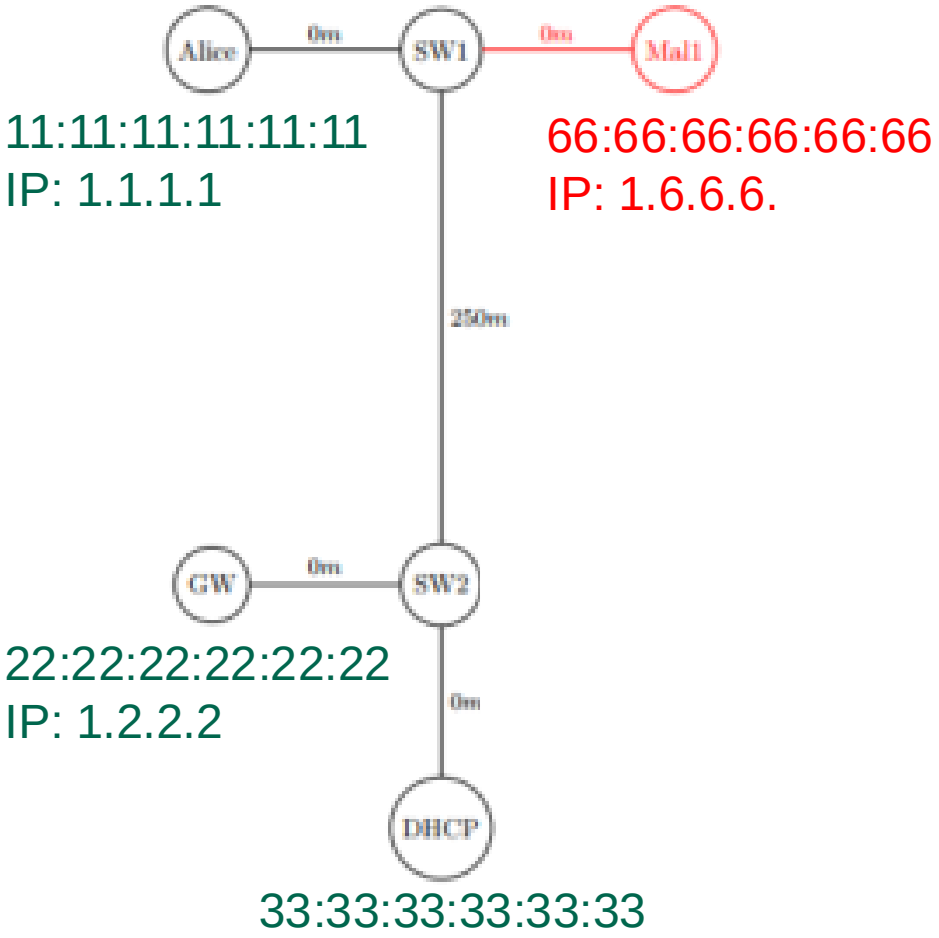
Exercise: ARP Poisoning MitM (3)

- Demonstrate MitM between Alice and GW (Internet), by spoofed ARP response
  - Assume Alice sends ARP request



Exercise: ARP Poisoning MitM (4)

- Demonstrate MitM between Alice and GW (Internet), by spoofed ARP response
  - With only Mal1!
  - Hints:
    - Using Ethernet
    - Poisoning with 'good' probability suffices



Preventing `MITM via ARP Poisoning`

Host-based defenses:

- Static address resolution tables (IP→MAC)
- Ignore unsolicited mappings (in request, response)
  - Broadcast new ARP request? Remove entry?
    - Overhead!!
- Re-do ARP requests
  - Upon detecting poisoning of self
  - Upon no response from peer [+ periodic exchanges]

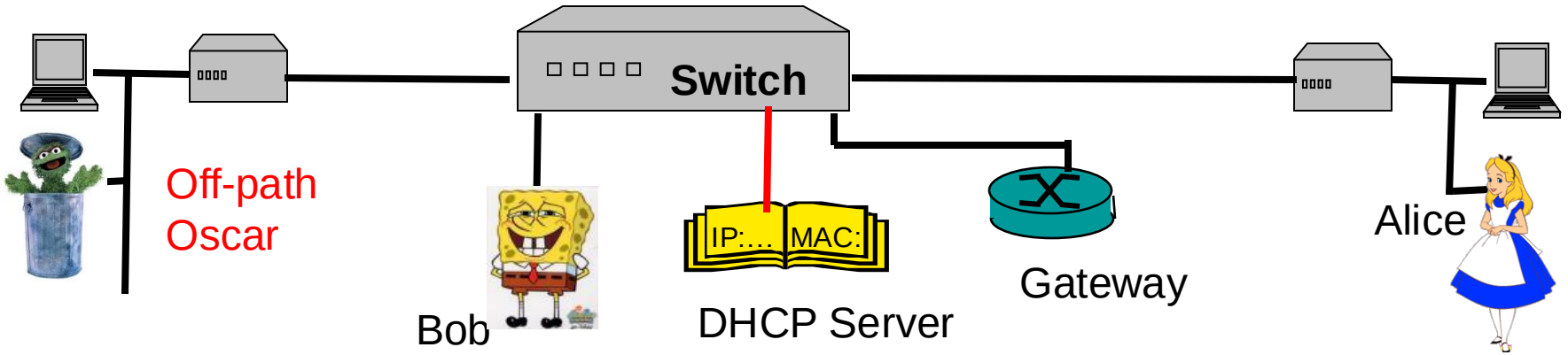
Network-based defenses:

- Monitoring to detect ARP-poisoning packets, MAC
- DHCP-authenticated mapping

Switch-based Port Security…

Switch Port-Security Defenses

- Detect then Disconnect, Alert, Correct/Prevent
- Multiple MAC/IP addresses from same port
- Excessive ARP requests/responses from port
- ARP messages conflicting with DHCP
- DHCP responses (except from DHCP port)



Thank you !

End of
(LAN) Security Lecture