# CS 4457 - Networks Lecture 6 Networking Layer Part I

Nate Hunter, Andrea Zhang

January 31, 2020

## Part I
# VLANs: Virtual Local Area Network

## 1 How do VLANs work?

- Switches that supports VLANs allow multiple virtual local area networks to be defined over a single physical local area network infrastructure.
- In a port-based VLAN, the switch's ports (interfaces) are divided into groups by the switch management software so that single physical switch operates as multiple virtual switches.

## 2 Port-based VLANS advantages

### 2.1 traffic isolation

If we define VLAN based on switch ports, and let Port 1 to 8 belongs to one VLAN, frames to/from ports 1-8 can only reach ports 1-8. We can also define VLAN based on MAC addresses of endpoints.

### 2.2 dynamic membership

Ports can be dynamically assigned among VLANs. The network manager declares a port to belong to a given VLAN (with undeclared ports belonging to a default VLAN) using switch management software, a table of port-to-VLAN mappings is maintained within the switch.

### 2.3 forwarding between VLANS

#### 2.3.1 Router

Suppose there are two VLANS A and B that share the same physical switch. We can connect a VLAN switch port to an external router and configure that port to belong both the VLAN A and VLAN B. In this case, the logical configuration would look as if A and B had separate switches connected via a router. An IP datagram going from A to B would first cross A to reach the router and then be forwarded by the router back over B to B's host.

#### 2.3.2 Switches plus routers

Switch vendors make such configurations easy for the network manager by building a single device that contains both a VLAN switch and a router, so a separate external router is not needed.

# 3 VLANS spanning multiple switches

A scalable approach to interconnecting VLAN switches is known as **VLAN trunking**. **Trunk Port** carries frames between VLANS defined over multiple physical switches. The trunk port belongs to all VLANs, and frames sent to any VLAN are forwarded over the trunk link to the other switch.

**Q:** How does a switch know that a frame arriving on a trunk port belongs to a particular VLAN?

**A:** The IEEE has defined an extended Ethernet frame format, 802.1Q, for frames crossing a VLAN trunk. The 802.1Q frame consists of the standard Ethernet frame with a four-byte VLAN tag added into the header that carries the identity of the VLAN to which the frame belongs. The VLAN tag is added into a frame by the switch at the sending side of a VLAN trunk, parsed, and removed by the switch at the receiving side of the trunk.

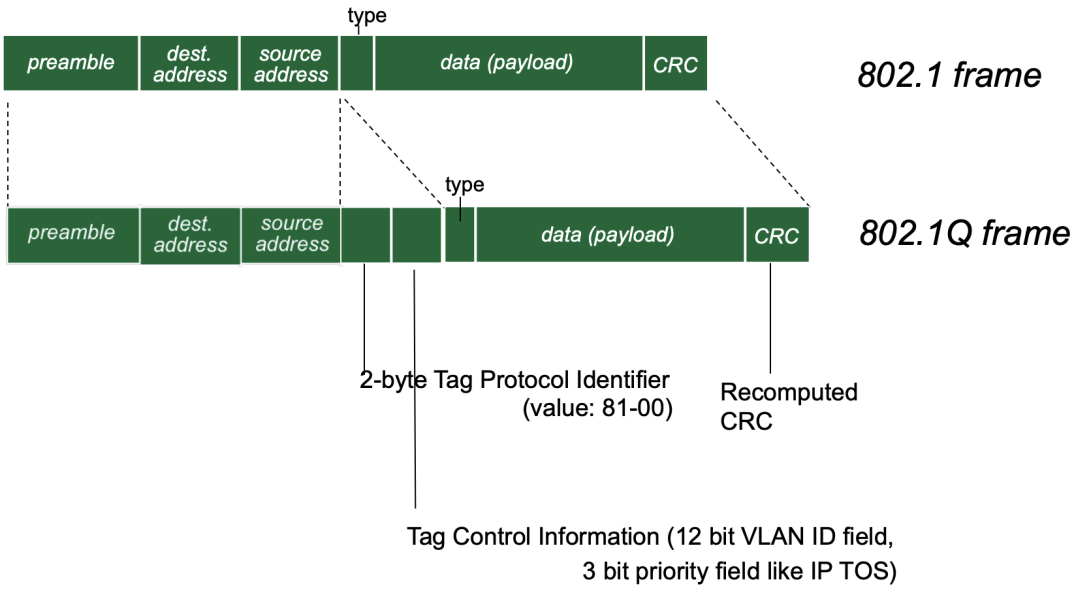


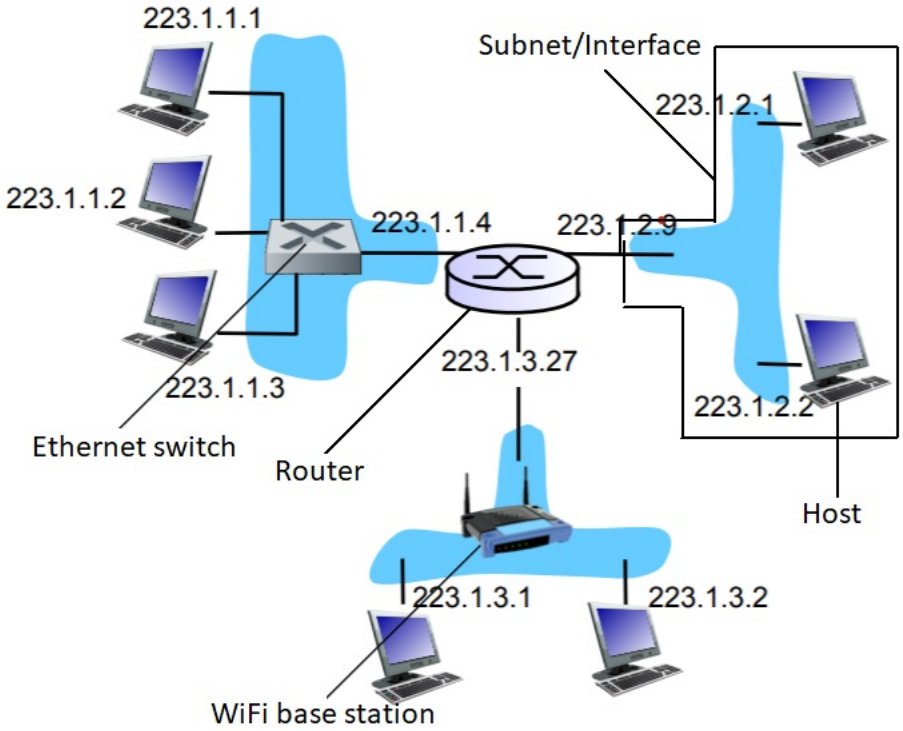

Figure 1: 802.1Q VLAN frame format

## Part II

# IPv4 addressing

An IP address is a 32-bit identifier for a host/router interface, essentially serving the same function as a mailing address for Internet packets. An example IP address would be:
192.168.1.10 = [11000000 10101000 00000001 00001010].

An interface is a connection between a router and a host (or another router) over some physical link. Hosts (such as your laptop) typically have 1 or 2 interfaces (Ethernet and 802.11/WiFi). Routers have multiple interfaces, as they serve to connect smaller sub-networks (single interfaces) together into a larger network. The internal connections of an interface are managed by a switch, such as an Ethernet switch or a WiFi base station.




A subnet is a part of a network that is connected by one interface, thus requiring no intervening router. This can look like several hosts connected by a switch or even just one wire between two routers. IP addresses thus consist of a subnet part (the higher-order bits common to all devices on the subnet) and a host part (the lower-order bits unique to each device). In the image above, the subnet part of the IP address for the boxed-out subnet is 223.1.2 (the first 24 bits).

Classless interdomain routing (CIDR) describes how IP addresses are allocated to these subnets. Such addresses take the form a.b.c.d/x, where x is the number of bits in the subnet part. In the example above, this would look like 223.1.2.2/24, as there are 24 bits specifying this particular subnet. A subnet mask can be used to obtain the subnet part of this address. Such a mask would look like 255.255.255.0, which is all 1's for the first 24 bits and all 0's for the last 8. By employing a bitwise AND between the subnet mask and the IP address, one arrives at just the subnet part of the IP address.

# Part III
# DHCP: Dynamic Host Configuration Protocol

DHCP is a client-server protocol. A client is typically a newly arriving host wanting to obtain network configuration information, including an IP address for itself. In the simplest case, each subnet will have a DHCP server. If no server is present on the subnet, a router that knows the address of a DHCP server for that network is needed.

## 3.1 Goals

- Allow host to dynamically obtain its IP address from network server when it joins network
- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

## 3.2 A four-step process

1. DHCP server discovery

   The newly arrived host wants to find a DHCP server with which to interact by using a **DHCP Discover message**. Since the host doesn't know the IP address of the DHCP server, it creates an IP datagram containing its DHCP discover message along with the broadcast destination IP address of 255.255.255.255 and a "this host" source IP address of 0.0.0.0. The DHCP client passes the IP datagram to the link layer, which then broadcasts this frame to all nodes attached to the subnet.

2. DHCP server offer(s)

   A DHCP server receiving a DHCP discover message responds to the client with a DHCP offer message that is broadcast to all nodes on the subnet. The DHCP server broadcasts the reply because several servers can be present in the subnet and the client may choose among several offers.

3. DHCP request

   The newly arriving client will choose from among one or more server offers and respond to its selected offer with a DHCP request message, echoing back the configuration parameters.

4. DHCP ACK

   The server responds to the DHCP request message with a DHCP ACK message, confirming the requested parameters.

Once the client receives the DHCP ACK, the interaction is complete and the client can use the DHCP-allocated IP address for the lease duration. Note the first two steps are optional.

### 3.3 DHCP client-server scenario

Figure 2 and 3 show how the DHCP works.

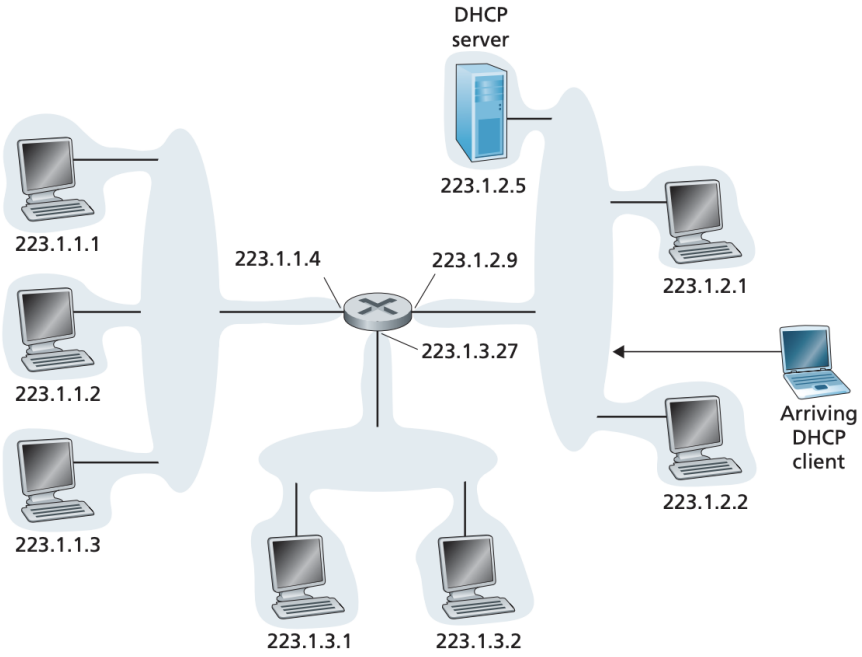



Figure 2: DHCP client-server scenario

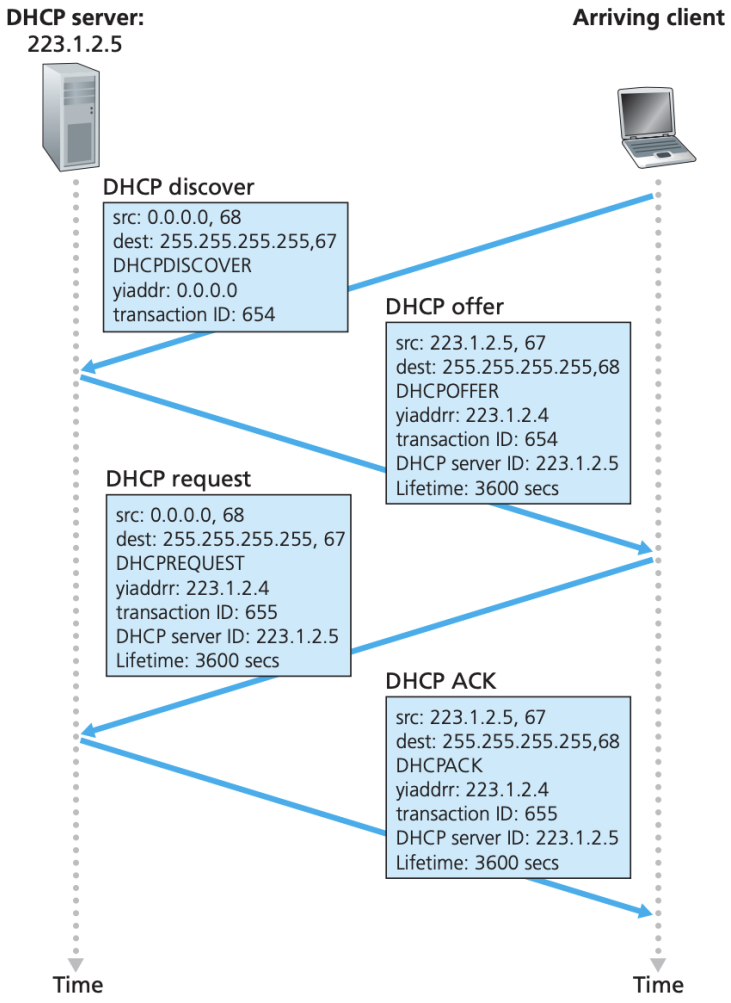



Figure 3: DHCP client-server interaction

# 4 DHCP Example

1. The new connecting laptop needs its IP address, address of first-hop router and address of DNS. The server decideds to use the DHCP.
2. DHCP request is encapsulated in UDP, then encapsulated in IP, and then encapsulated in 802.1 Ethernet.
3. Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN. It is received at router running DHCP server.
4. Ethernet demuxed to IP demuxed, UDP demuxed to DHCP.
5. DCP server formulates DHCP ACK containing client's IP address, IP address of firsthop router for client, name & IP address of DNS server.
6. encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client.

7. client now knows its IP address, name and IP address of DSN server, IP address of its firsthop router.

# Part IV
# Hierarchical addressing

A hierarchical addressing scheme that breaks the address space up into ordered chunks. Telephone numbers are a great example of this type of addressing. The first section of a telephone number, the area code, designates a very large area; the area code is followed by the prefix, which narrows the scope to a local calling area. The final segment, the customer number, zooms in on the specific connection.

IP addressing works the same way. Instead of treating the entire 32 bits as a unique identifier, one part of the IP address is designated as the network address (or network ID) and the other part as a node address (or host ID), giving it a layered, hierarchical structure.

The network address uniquely identifies each network. As figure 4 shows, every machine on the same network shares that network address as part of its IP address, just as the address of every house on a street shares the same street name. In the IP address 130.57.30.56, for example, 130.57 is the network address.

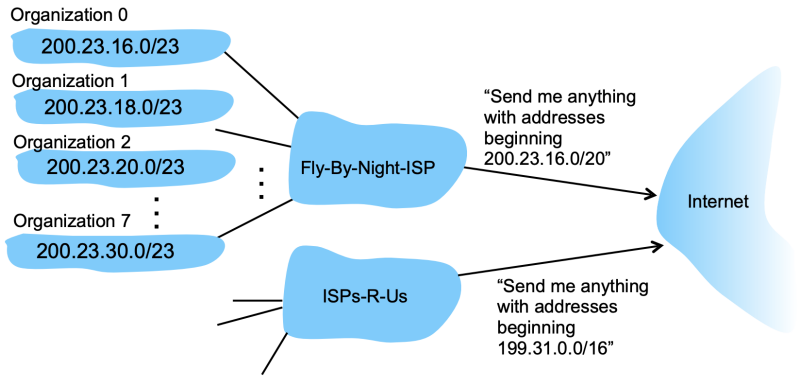


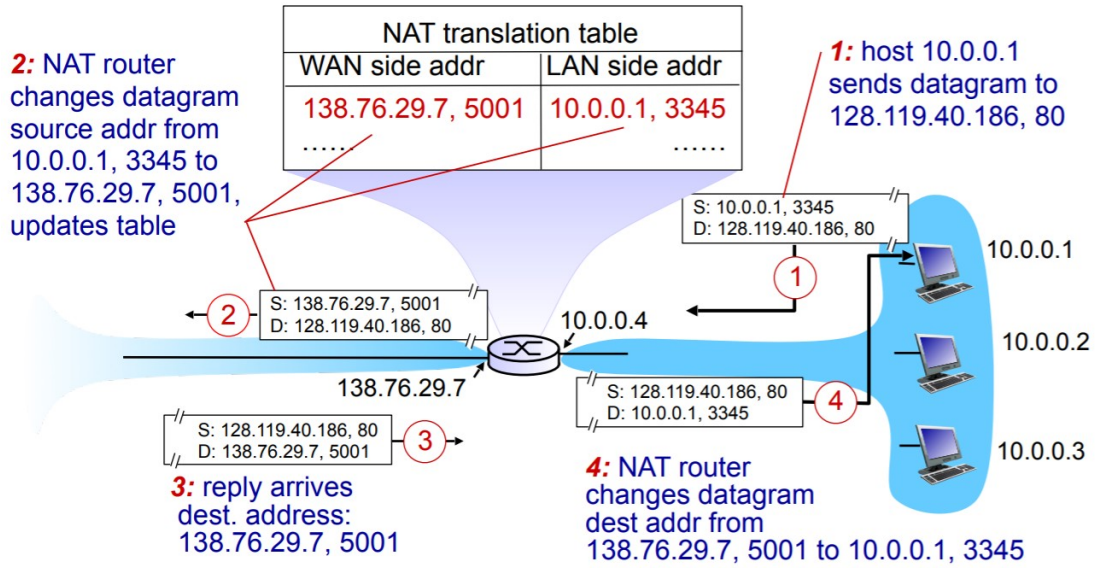

Figure 4: Hierarchical Addressing

## Part V

# NAT: network address translation

In IPv4, addresses are only 32 bits, so there are $2^{32} \approx 4$ billion possible addresses. This is less than the population of the world. One solution, so that everyone can still have a unique identifier, is to use network address translation (NAT).

In network address translation, every LAN has just one IP address to the outside world. Thus, in addition to the benefit of increasing the address space, NAT also serves to isolate changes in the WAN from changes in the LAN. Addresses can change inside the LAN without notifying the Internet, and the ISP can change without affecting the device addresses in the LAN. Furthermore, devices in the LAN are not explicitly addressable/visible to the outside world, providing security. So in short, NAT has a lot of benefits.

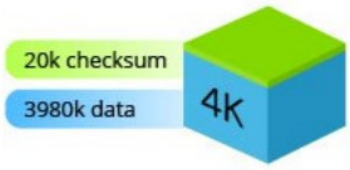



The diagram above shows how NAT works. When a host sends a datagram to the Internet (1), the NAT router first changes the host's LAN address and port number to a WAN address and port number and adds this information to its NAT table (2). When the reply comes back to the NAT router from the Internet (3), the router refers to the table to convert the received WAN address and port number back into the host's LAN address and port number (4). Ultimately, NAT expands the address space by utilizing the 16-bit port identifier, allowing $2^{16} = 65,536$ unique devices to be identified for every one LAN.

## Part VI

# IP fragmentation, reassembly

Network links have a maximum transmission unit (MTU), which specifies the maximum length of a packet on that link. This length varies from link to link, so to get larger packets through smaller links, it is sometimes necessary to fragment the packet into smaller sub-packets and then to reassemble the sub-packets into the larger packet at the other end. This can be accomplished by maintaining ordering information in the IP packet headers (in the offset field). The figure below shows the breakdown of a potential fragmentation; note the added checksum data for each packet.




| length | ID | fragflag | offset |
|---|---|---|---|
| 4000 | X | 0 | 0 |

**MTU = 1500 Bytes, Offset = 1480/8**


| length | ID | fragflag | offset |
|---|---|---|---|
| 1500 | X | 1 | 0 |


| length | ID | fragflag | offset |
|---|---|---|---|
| 1500 | X | 1 | 185 |


| length | ID | fragflag | offset |
|---|---|---|---|
| 1040 | X | 0 | 370 |

# Part VII
# IPv6

An additional solution to the address space limitations of IPv4 is IPv6, which uses a longer IP address. IPv6 also has benefits of a more efficient header format, which can speed up processing/forwarding and facilitate QoS (quality of service - the enforcement of specific bandwidths for particular services). Following are some of the features of IPv6 and a diagram of its datagram format.

- Address expanded to 128 bits
- Fixed-length 40-bit header
- No fragmentation (secure against fragmentation attacks)
- Priority: prioritizes datagrams in flow
- Flow label: identifies datagrams in same flow (concept not well-defined)
- Next header: identifies upper layer protocol