

Lecture 2



Picking up from Lecture 1

- Chapter 1



What is Cloud Computing?

- IBM (2016) defines cloud computing as follows:
“Cloud computing, often referred to as simply “the cloud,” is the delivery of on-demand computing resources—everything from applications to data centers—over the Internet on a pay-for-use basis.”
- Applications running on the cloud often employ a service oriented architecture



What is Cloud Computing?

- Public cloud:
 - instead of doing your own computing and storing your data on the computer on your desktop yourself
 - you hire a company to do the computing and store the data on their big computer servers that you access via the web
- Private cloud:
 - instead of having computing and data storage on employees' desks
 - the company can buy its own big servers
 - then the employees do their computing and store their data on those servers
- Hybrid cloud:
 - Part of a company's computing is done in house
 - Part of a company's computing is done by a public cloud



What is Cloud Computing?

- There are three different paradigms for cloud computing:
 - Infrastructure as a Service (IaaS)
 - you or your company pays a cloud provider for computing resources
 - You provide your own operating system and application software
 - Platform as a Service (PaaS)
 - you or your company pays a cloud provider for an environment provided by the company that provides everything you need in order to develop and run your applications
 - This environment includes operating system, development tools, web site hosting, among other things
 - Software as a Service (SaaS)
 - you or your company pays a cloud provider for the use of their software application



Chapter 2

- Design patterns for cloud.



Architectural Styles/Patterns for Web Services

- RESTful vs. non-RESTful
- We will examine this in depth later on
- However, briefly:
 - you can (sort of) think of a non-RESTful web service as being yet another example of a remote procedure call
 - RESTful web services, however, use the HTTP protocol layer directly and employ HTTP messages such as GET, POST, etc.,
 - thus they do not follow the remote procedure call paradigm

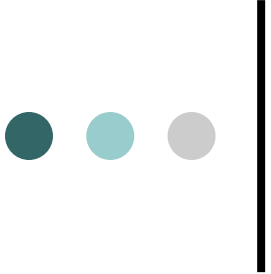


Architectural Styles/Patterns for Cloud Computing—Cloud Communication

Melendez et al. (2015) looks at communication between a cloud client and a cloud service provider.

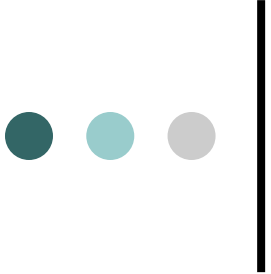
They divide communication between cloud clients and cloud service providers into categories based on the amount of user-interaction involved:

- interactive
- non-interactive



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

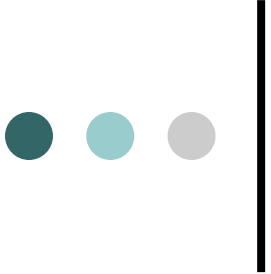
- Software as a Service (SaaS) tends to be interactive
 - uses a thin client (small client with most processing done on the server)
- Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) are often *non* interactive
 - offload a computational job to the cloud



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

Interactive communication (cont'd)

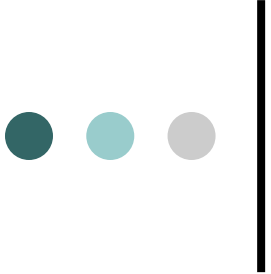
- According to Melendez et al. (2015) interactive communications results in an asymmetric traffic pattern
 - Since data from the user (working on the thin client) to the cloud service provider is typically in much smaller quantities than from the cloud service provider to the user
 - So the network throughput from the cloud to the user should be higher



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

Interactive communication

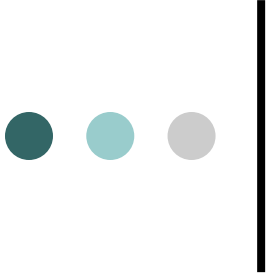
- a user sends data to the cloud and receives data from the cloud in an interactive manner
- because a user must wait for a response from the cloud, the latency of the communication is important
 - According to Melendez et al. (2015), based on Human Computer Interaction Studies, response times can be categorized as follows:
 - Not noticeable: <150 milliseconds
 - Acceptable: between 150 and 400 milliseconds
 - Unacceptable: > 400 milliseconds



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

Noninteractive communication

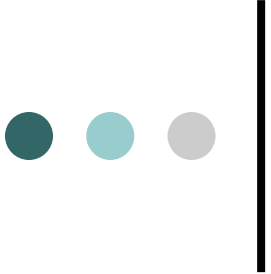
- often includes large file transfers
 - Application program and/or input data and/or output data may be uploaded to cloud or downloaded from cloud
 - It is assumed that the actual execution of the application program will take place on the cloud



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

Melendez et al. (2015) look at communication within a cloud data center. It can be divided into:

- north-south traffic
 - Traffic that leaves the data center (from client to/from cloud service provider)
- east-west traffic
 - Traffic that stays inside the data center



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

The quantity of north-south traffic in a particular data center compared to the quantity of east-west traffic depends on the applications that use that data center.

- For example, data mining applications are mostly east-west.



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

A typical physical data center architecture consists of a rack of servers and storage devices

- with a switch at the top of the rack

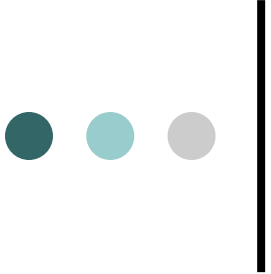
The racks are aggregated into groups, each with its own switch

- the groups are aggregated into the main data center router, which connects the cloud to the internet



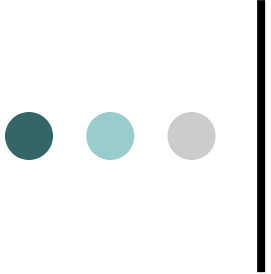
Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

- Based on previous traffic studies, east-west traffic primarily stays within a single rack
- most communication occurs among adjacent servers
 - the magnitude of traffic between servers decreases as the physical distance between the servers increases



Architectural Styles/Patterns for Cloud Computing—Cloud Communication (cont'd)

- A server is typically either a receiver of data or a sender of data.
 - However, if a server is primarily a receiver, it will send back an almost equal number of data packets in the form of acknowledgement messages.



Architectural Styles/Patterns for Cloud Computing—Microsoft Cloud Design Patterns

- Homer et al. (2014) from Microsoft provide twenty-four design patterns that they feel can be helpful in an application that is hosted in a cloud
 - Note that they are not specifically talking about how the cloud itself is implemented.



Architectural Styles/Patterns for Cloud Computing—Microsoft Cloud Design Patterns (cont'd)

They divide their patterns into categories that are based on the main problem areas in application development for a cloud. These categories are:

- Availability—the amount of time the application is available
- Data Management—how can data hosted at different locations be kept synchronized
- Design and implementation—consistency, maintainability, reusability, etc.
- Messaging—asynchronous messaging is widely used because of the need for loose coupling between services
- Management and monitoring—how does monitoring occur in a remote data center?



Architectural Styles/Patterns for Cloud Computing—Microsoft Cloud Design Patterns (cont'd)

- Performance and scalability—throughput, response time (latency), scalability (ability to handle increases in workload)
- Resiliency—ability to gracefully recover from a failure
- Security—prevent malicious use of cloud resources. Preserve user data privacy.



Architectural Styles/Patterns for Cloud Computing—Microsoft Cloud Design Patterns (cont'd)

The messaging patterns include:

- Competing Consumers—different servants receive messages on the same channel
- Priority queue—messages to servants are prioritized
- Queue-based load leveling—a service queues up tasks, so that intermittently heavy loads can be better handled
- Scheduler agent supervisor—coordinate actions across a distributed set of resources, be able to roll back the set of actions if necessary



Architectural Styles/Patterns for Cloud Computing—Microsoft Cloud Design Patterns (cont'd)

The performance/scalability patterns include (among others):

- Competing Consumers—see above
- Command and Query Responsibility Segregation—use a separate interface to read data and a separate interface to update data
- Queue-based load leveling—see above
- Throttling—control resource consumption so that demand from one or more entities doesn't overrun resources



Architectural Styles/Patterns for Cloud Computing—Microsoft Cloud Design Patterns (cont'd)

The security patterns include (among others)

- Federated identity—use an external identity provider to perform authentication
- Valet key—instead of a client having to re-authenticate (user name/password, etc.) every time the client needs to access a resource, the application does authentication once and then provides the client with a time limited token. The client then uses the token to access the resource.



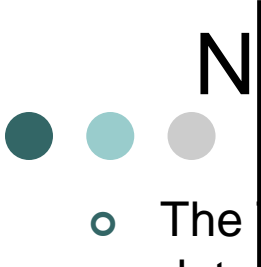
Chapter 3

- Networking basics.



Internet Technologies

- The internet uses a protocol suite known as TCP/IP:
 - where TCP stands for Transmission Control Protocol
 - IP stands for Internet Protocol
- TCP/IP was created back in the early to mid-1970s and was formally adopted by ARPANET in 1983



Network Basics—Transmission Control Protocol (TCP)

- The Transmission Control Protocol (TCP) is responsible for bidirectional data transmissions between two distributed applications on a network:
 - breaks data streams into data chunks (known as packets) at the sender
 - makes sure packets haven't been mangled during transmission and causing any packets with problems to be re-transmitted:
 - it does this by using a checksum field in the packet
 - if the checksum that was sent doesn't match the newly calculated checksum at the recipient, then the data in the packet was mangled
 - makes sure packets are reassembled in the correct order so that the data stream arrives correctly at the recipient:
 - it does this by numbering each packet. Then if packets arrive out of order, they can be rearranged before being sent up to the application.
- in order to do these tasks, TCP creates a connection (session) between the sender and the receiver



Network Basics—Internet Protocol

- Internet Protocol (IP) is responsible for actually making sure that individual packets get from the sender to a receiver:
 - it does this based on IP addresses in the IP packet header
 - each IP packet is routed, based on its header, across the internet from the sender to the receiver.
 - If some portion of the internet should require using packets that are smaller than the originally sent packet, then IP can break the bigger packets into smaller packets in order to get across that internet portion,
 - then reassemble them into the original packet after all the smaller packets get through that portion of the internet

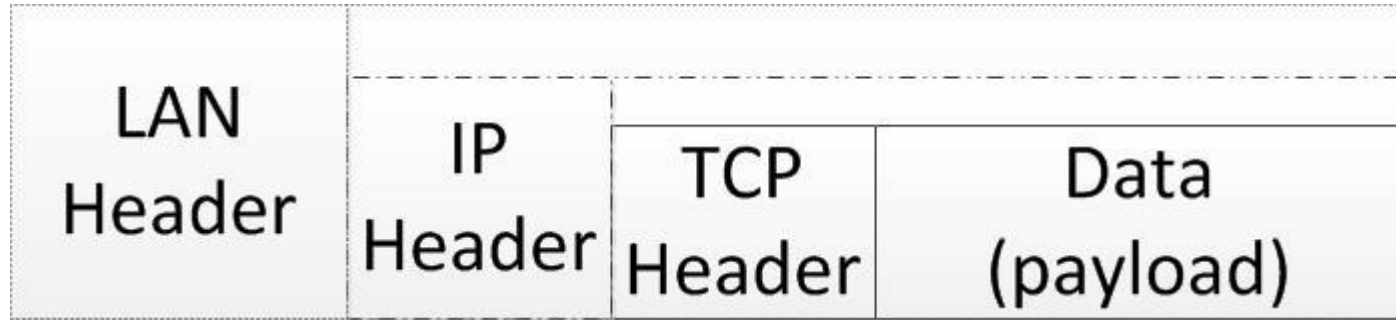
● Network Basics—Internet Protocol (cont'd)

- with an IP packet you don't know whether the receiver ever got the packet or not.
 - Also, the receiver doesn't know whether or not to expect any packets
- IP leaves those tasks for TCP to do (TCP does those tasks by establishing a connection)
- So IP taken by itself is considered to be a connectionless protocol

● Network Basics—Internet Protocol (cont'd)

- TCP is nested inside IP:
 - an entire TCP packet is encapsulated inside an IP packet
 - the IP packet is responsible for delivering its data to the recipient
- the entire IP packet is nested inside the local area network (LAN) frame (the LAN could be using Ethernet, for example):
 - the IP packet is nested inside the Ethernet frame when the packet is being sent across an Ethernet LAN
 - after the recipient acquires the frame from the LAN it will strip off the LAN headers.
 - then the recipient has to
 - interpret and handle the IP packet headers
 - and then extract the data from the IP packet before the TCP packet can be interpreted

● ● Network Basics—Internet Protocol (cont'd)



● ● Network Basics—Internet Protocol (cont'd)

- Another way of thinking of this is that the TCP packet layer lies on top of the IP packet layer.
 - This is generally how the OSI seven layer model works. In the OSI seven layer model:
 - it can be said that IP is at the network layer, and TCP is at the transport layer.
 - UDP is at the transport layer.
 - Frenzel (2013) provides a discussion of ways in which the TCP/IP protocol stack does not map well otherwise into the OSI seven layer model

● Network Basics—Internet Protocol (cont'd)

To establish a connection between two distributed entities, TCP does a three way handshake:

1. Host A asks Host B if it is willing to establish a connection
 - It does this by sending a synchronize (SYN) packet to Host B
2. Host B tells Host A that yes, it is willing
 - It does this by sending an I am willing packet (Synchronize Acknowledgement, or SYN-ACK) to Host A
3. Host A tells Host B well, ok, then, I'm ready to get going
 - It does this by sending an okay then packet (Acknowledgement or ACK) to Host B

● Network Basics—Internet Protocol (cont'd)

- To make sure that individual packets get from the sender to a receiver:
 - IP includes a source address and a destination address in its header.
 - when passing through the network, an IP packet normally must pass through several IP routers
 - the routers look at the IP address in the packet and forward that packet on to a computer (or to another router) based on the IP address

● ● Network Basics—Internet Protocol (cont'd)

- port numbers are used inside the TCP header to make sure that the data inside the TCP packets gets to the correct application
- the IP addresses get the data to the correct computer
- the port numbers get the data to the correct application on that computer

● ● Network Basics—User Datagram Protocol

- The User Datagram Protocol (UDP) is an alternative to TCP, both ride on top of IP
- UDP uses datagram service
 - it doesn't specify a connection
- it provides a checksum to make sure it's known whether or not the packet has been garbled during transmission
 - although it does not cause a retransmission if the packet is garbled.
- it includes port numbers so the data can get to the correct application on the receiving computer



Network Basics—IP Addresses

- TCP/IP defines a communication endpoint to consist of an IP address and a protocol port number
- There are two versions of IP on the internet,
 - the old version IP version 4, usually known as IPv4
 - and the newer IP version 6, usually known as IPv6

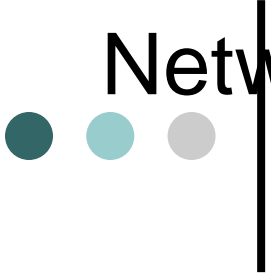


Network Basics—IPv4 Addresses

- An IPv4 address is 32 bits long, and is normally divided into four bytes
- When each byte is represented as a decimal number that is called “dotted decimal notation”
- For example, a typical IP address could be:
 - 100010110101001000010000010000
- When this is broken into four bytes, you get:
 - 01000101 01010100 10000100 00010000
- When each byte is represented as a decimal number, you get dotted decimal notation:
- 69.84.132.16

● ● Network Basics—IPv4 Addresses (cont'd)

- A port number can be appended to an IP address, let's append port number 1132:
 - 69.84.132.16: 1132
- A few special IP addresses:
 - 127.0.0.1—the loopback address, accesses your current computer
 - 224.0.0.1—multicast address, addresses all hosts on the same network segment
 - only devices which are members of the particular multicast group will accept packets from this address
 - 224.0.0.2—multicast address, addresses all routers on the same network segment
 - 255.255.255.255—broadcast address of the zero network (0.0.0.0)
 - zero network means the local network (this one is not forwarded by routers)



Network Basics—IPv4 Addresses (cont'd)— Private IP Addresses

- RFC 1918 reserves the following ranges of IP addresses as private addresses that cannot be routed on the Internet:
 - 10.0.0.0 - 10.255.255.255 (10/8 prefix)
 - 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
 - 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Network Basics—IPv4 Addresses (cont'd)—

Network Address Translation (NAT)

- Normally private IPs are mapped to one or more public IPs through the use of Network Address Translation (NAT)
- Network Address Translation allows a single device, perhaps a router, to interface between the outside internet and a local (private) network
 - in this way, only a single public IP address is required to represent an entire group of computers
 - internal to the private network, those computers employ private IP addresses, which are then mapped to a public IP by the router



Network Basics—IPv6 Addresses

- Why was IPv6 necessary? Bradley (2012) says:

“The most obvious answer is that IPv4 is out of IP addresses. IPv4 has only 4.3 billion addresses, and with PCs, smartphones, tablets, gaming systems, and just about everything else connecting to the Internet we've tapped the system dry. IPv6 uses 128-bit addresses and is capable of 340 undecillion addresses. That is 340 times 10 to the 36th power, or 340 trillion trillion trillion possible IP addresses.”
- IPv4 and IPv6 can run beside each other
- there are so many IPv4 addresses in use that it wasn't possible to ditch them in order to go straight to IPv6

● ● Network Basics—IPv6 Addresses (cont'd)

- IPv6 addresses use 128 bits divided into eight sets of four hexadecimal digits.
- An example of an IPv6 address:
 - 3001:0B0E:3247:CEF1:0073:0000:0000:AF0C
- IPv6 shorthand notation where leading zeros are dropped. For example:
 - 3001:0B0E:3247:CEF1:**73**:0000:0000:AF0C
 - notice the “73” in italics in the middle, 5th set of 4 hex digits.
 - this is shorthand notation for “0073”

● ● Network Basics—IPv6 Addresses (cont'd)

- If at least two consecutive sets of 4 hex digits have the hex digits all zero, then a shorthand notation which replaces those with a “:” can be used:
 - 3001:0B0E:3247:CEF1:73::AF0C
 - note that here there is a “::” just before AF0C
 - this double colon represents two sets of 4 hex digits (8 hex digits total)

● Network Basics—IPv6 Addresses (cont'd)

- The four sets of 4 hex digits in the following example:
 - 3001: 0B0E:0000:0000:0000:0000:32AF: AF0C
- can be replaced by a double colon:
 - 3001: 0B0E::32AF:AF0C
- you can only use “::” once in an address
- you represent an unspecified address as “::”
 - it's all zeroes

● ● Network Basics—IPv6 Addresses (cont'd)

- A few special IP addresses:

- 0000:0000:0000:0000:0000:0000:0000:0000

- IPv6 loopback address

- also shown as ::1

- IPv6 loopback address

- ff02::1

- multicast, addresses all hosts on the local network segment

- note that all multicast addresses start with 0xFF



Network Basics—Subnetting

- Networks can be broken into chunks called subnets, based on their IP addresses.
- When you're using a subnet, an IP address is considered to consist of a Network Prefix followed by a Host address
 - it's the same format IP address as before, you just pick several of the most significant bits and call them a “network prefix”
 - the remaining bits will be the host address



Network Basics—Subnetting

- the IP addresses of a network can be broken into chunks called subnets
- with a subnet, an IP address consists of:
 - network prefix
 - host address
 - it's the same IP address as before, you just pick several of the most significant bits and call them a “network prefix,” the remainder will be the “host address”



Network Basics—Subnetting

Classless Inter-Domain Routing (CIDR) format looks like the following:

- 69.84.132.16/16
- here the first 16 bits of the IP address are used to specify the Network Prefix

A subnet mask that has 1s in every location in the first 16 bits would be ANDed with the raw IP address to extract the Network prefix:

- the subnet mask corresponding to this would be (in dotted decimal notation):
 - 255.255.0.0
- which corresponds to, in binary:
 - 11111111 11111111 00000000 00000000



Network Basics—Subnetting

The binary address that corresponds to the dotted decimal notation 69.84.132.16 is:

- 01000101 01010100 10000100 00010000

So this number ANDed with the subnet mask would be:

- 01000101 01010100 00000000 00000000

or in dotted decimal notation:

- 69.84.0.0

- which is the network prefix



Network Basics—Subnetting

We used an even number of bytes in the CIDR format example:

- 69.84.132.16/16

You don't have to use an even number of bytes, for example, the following would be acceptable:

- 69.84.132.16/12

For this one, the subnet mask would be:

- 11111111 11110000 00000000 00000000

or in dotted decimal notation:

- 255.240.0.0

So if you AND this with the raw IP address (69.84.132.16) you get:

- 69.80.0.0



Network Basics—Subnetting

- A router uses a subnet mask ANDed with an IP address to extract the Network Prefix
- The Network prefix is then used by a router to route a packet to the appropriate subnet
- The remaining host address can be used to route the packet to the appropriate host within that subnet



Network Basics—Subnetting

- When using IPv6, the smallest recommended subnet is 64 bits (of its total 128 bit wide address) as the Network Prefix
- So the 64 bit prefix is always used for a subnet, instead of having a variable prefix
- You're not supposed to use a subnet smaller than that
 - that is, using more bits as the Network Prefix, and thus having fewer host addresses



Network Basics—Subnetting

According to RIPE Network Coordination Centre (2015):

“Currently, most ISPs assign /48 network prefixes to subscribers’ sites (the End Users’ networks). Because all IPv6 networks have /64 prefixes, a /48 network prefix allows 65,536 LANs in an End User’s site.”



Network Basics—Port Numbers

- An internet port number is part of an address
- A server listens for input on a particular port number
- Port numbers range from 0 to 65535
 - Port numbers in the range 0 to 1023 are called “well known ports” or “system ports,”
 - preassigned by the Internet Assigned Numbers Authority, IANA (2016), to certain functions (by default)



Network Basics—Port Numbers

- http accesses port number 80 by default
 - so `http://myownplace/MyService`
 - defaults to port 80
- https accesses port number 443 by default
- You can use other port numbers with http or with https if you specify them
- Port numbers work the same with IPv6 as with IPv4



● ● ● Internet Control Message Protocol (ICMP)

- ICMP is used to send messages back to the source IP address in cases of error or some sort rerouting
 - a typical ICMP message is “Destination Unreachable”
- ICMP messages are sent as payload (data) inside IP packets
- A “ping” is an ICMP message containing an “echo request” header
 - The host responds with an “echo reply” header



Internet Control Message Protocol (ICMP)

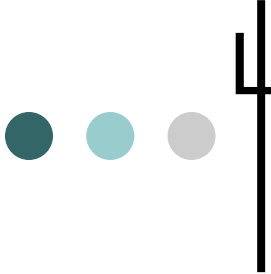
- ICMP message with “timestamp” header followed by an ICMP message with “timestamp reply” can be used to determine network delays
- However, according to Mitre (2016):

“An attacker may be able to use the timestamp returned from the target to attack time-based security algorithms, such as random number generators, or time-based authentication mechanisms.”



Local Area Networks (LANs)

- A local area network (LAN) traditionally is a computer network that connects computers that are located close to each other.
- Commonly used protocols for local area networks are Ethernet and wifi



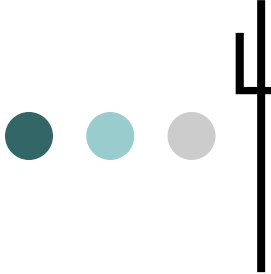
Local Area Networks (LANs)—Media Access Control (MAC) Layer

- A MAC Layer is a generic way to look at different LAN protocols
- MAC layers form data frames that go from one Network Interface Card (NIC) on a LAN to another Network Interface Card (NIC) on a LAN
 - A bunch of data is called a “frame” at the data link layer but it’s called a “packet” at higher layers



Local Area Networks (LANs)—Media Access Control (MAC) Addresses

- A Media Access Control (MAC) address is assigned to a physical device at the time the device is manufactured
- The Institute of Electrical and Electronics Engineers (IEEE) Registration Authority (RA) assigns blocks of MAC addresses to companies, in return for a fee



Local Area Networks (LANs)—Media Access Control (MAC) Addresses

Traditionally MAC addresses are 48 bits long in hexadecimal, with bytes divided by colons, as follows:

- xx: xx: xx: yy: yy: yy
 - where x represents hex digits assigned to a company (an Organizationally Unique Identifier, or OUI)
 - so the OUI is 24 bits long
 - the OUI is purchased by the company from the IEEE RA
 - the last several hex digits (y) are in a pool of addresses assigned to the company



Local Area Networks (LANs)—Media Access Control (MAC) Addresses

Today there are two kinds of MAC addresses

- 48-bit Extended Unique Identifier (EUI-48)
 - EUI-48 is a string of six bytes in hexadecimal format.
 - IEEE Standards Association (2016) gives the following example:
 - AC-DE-48-23-45-67
- 64-bit Extended Unique Identifier (EUI-64)
 - EUI-64 is a string of eight bytes in hexadecimal format.
 - IEEE Standards Association (2016) gives the following example:
 - AC-DE-48-23-45-67-AB-CD



Local Area Networks (LANs)—Media Access Control (MAC) Addresses

- The broadcast MAC for a 48 bit MAC is:
 - FF:FF:FF:FF:FF:FF
- A MAC address on a Virtual Machine is typically assigned by the hypervisor



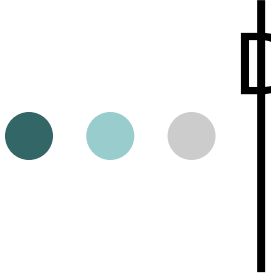
Hubs and Bridges and Switches and Routers

- A hub is a layer 1 (physical layer) device
 - When data comes in one port of a hub it is copied out all other ports of the hub
- A Bridge is a layer 2 device that connects one layer 2 network segment to another layer 2 network segment
 - A bridge looks at the MAC address in a frame
 - if the destination is not on the other side of the bridge it will not transmit the data across
 - If the destination is on the other side of the bridge then it will transmit the data across
 - A bridge is very simple, it is not a practical device when trying to connect more than two devices



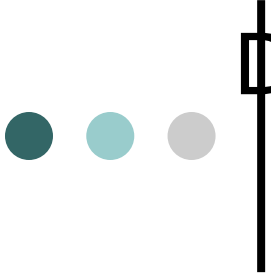
Hubs and Bridges and Switches and Routers

- A Switch is a layer 2 device that acts as a multiport bridge and maps MAC addresses to ports
 - A frame would come in on one port
 - The switch looks at the MAC address in that frame
 - It sends the frame to the port that is connected to that MAC address.
- A Router is a layer 3 device that connects multiple layer 3 networks and uses IP addressing
 - A broadcast domain is a network segment in which any network device can transmit data directly to another device without going through a router
 - So a router is the edge of a broadcast domain



Dynamic Host Configuration Protocol (DHCP) for IPv4

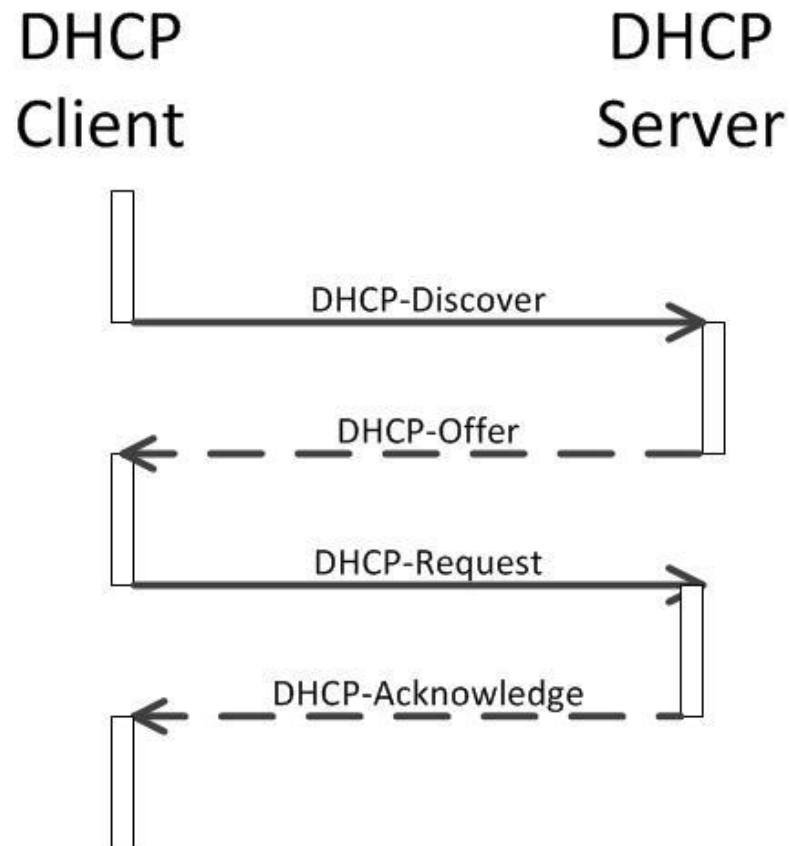
- Static IP—(without DHCP) a device on a LAN has its IP address manually configured by a sysadmin
 - If the device is moved to a new location on a different network (for ex., your laptop at Starbucks), it must be manually reconfigured
- DHCP performs autoconfiguration of:
 - IP addresses, but also perhaps:
 - subnet mask
 - default gateway
 - domain name

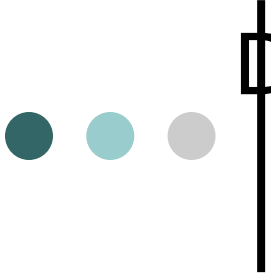


Dynamic Host Configuration Protocol (DHCP) for IPv4

- DHCP is connectionless
 - DHCP messages ride inside UDP packets
 - Port number for DHCP client is 68
 - Port number for DHCP server is 67
- A DHCP client requests an IP address from a DHCP server
 - DHCP server gives a lease to DHCP client for a period of time
 - At the end of this time the client must request a new IP address (or to be reassigned its old IP address)

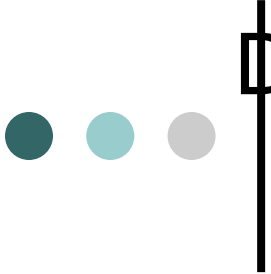
Dynamic Host Configuration Protocol (DHCP) IPv4 Handshake





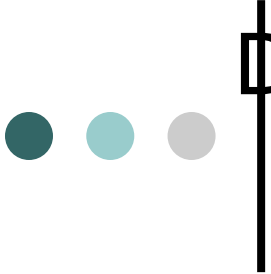
Dynamic Host Configuration Protocol (DHCP) for IPv4

- DHCP-Discover with IPv4
 - Source address 0.0.0.0
 - Destination address 255.255.255.255
 - Includes MAC address of client
- When DHCP-Server receives DHCP-Discover message
 - DHCP server has a pool of available IP addresses
 - Reserves an IP address
 - Sends DHCP-Offer message to broadcast 255.255.255.255
 - Includes MAC address of client
 - IP address being offered
 - Lease duration
 - server identification (in case multiple DHCP servers respond)



Dynamic Host Configuration Protocol (DHCP) for IPv4

- Client replies by sending DHCP-request
 - Source address 0.0.0.0
 - Destination address 255.255.255.255
 - Server identification
- Identified server knows that its offer was accepted
 - Sends back a DHCP-ACK
 - Destination address 255.255.255.255
 - Source address of DHCP server
- Servers that were not identified know their offer was not accepted



Dynamic Host Configuration Protocol (DHCP) for IPv4

- Linux can be configured to act as a DHCP for IPv4 Server
- Windows Server can be configured to act as a DHCP for IPv4 server
- Many routers can be configured to act as DHCP for IPv4 servers



Autoconfiguration for IPv6

- IP address configuration in IPv6 can be done several ways:
 - static addressing can still be used, only with IPv6 addresses instead of IPv4 addresses
 - static addressing—the IP address is assigned statically but other configuration information is assigned using (stateful) DHCP for IPv6
 - stateful autoconfiguration—the entire IP address is assigned and other configuration information is assigned using (stateful) DHCP for IPv6
 - Stateless Address Autoconfiguration (SLAAC)



Autoconfiguration for IPv6—SLAAC

- With SLAAC, an IPv6 network router sends out a Router Advertisement (RA) periodically
- The RA includes:
 - IPv6 subnet prefixes of length 64, that is /64
 - lifetime information for the prefix
 - default router to use
 - lifetime of default router



Autoconfiguration for IPv6—SLAAC

- With SLAAC, the device on the network that needs an IP address will use the IPv6 subnet prefix for the most significant 64 bits of the IPv6 address
- Then the bottom 64 bits of the IPv6 address are formed by using a manipulation of the device's MAC address using EUI-64 rules



Autoconfiguration for IPv6—SLAAC

The EUI-64 rules work as follows:

- To transform an IEEE EUI-64 identifier to an interface identifier all that must be done is to invert the "u" (universal/local) bit
 - The universal/local bit is the seventh bit, counting from the top of the word
- To transform an IEEE EUI-48 identifier to an interface identifier
 - insert the hexadecimal value 0xFFFE in between the Organizationally Unique Identifier (OUI) in the 24 bits at the top of the 48 bit MAC address, and the manufacturer supplied identifier in the bottom 24 bits of the 48 bit MAC address
 - This makes it a 64 bit identifier (48bits plus 16 additional bits equals 64 bits)
 - Then invert the universal/local bit, which is the seventh bit, counting from the top of the word



Autoconfiguration for IPv6—SLAAC

The EUI-64 rules work as follows:

- To transform an IEEE EUI-64 identifier to an interface identifier all that must be done is to invert the "u" (universal/local) bit
 - The universal/local bit is the seventh bit, counting from the top of the word
- To transform an IEEE EUI-48 identifier to an interface identifier
 - insert the hexadecimal value 0xFFFE in between the Organizationally Unique Identifier (OUI) in the 24 bits at the top of the 48 bit MAC address, and the manufacturer supplied identifier in the bottom 24 bits of the 48 bit MAC address
 - This makes it a 64 bit identifier (48bits plus 16 additional bits equals 64 bits)
 - Then invert the universal/local bit, which is the seventh bit, counting from the top of the word



Autoconfiguration for IPv6—SLAAC

There are privacy considerations with SLAAC. For example:

- Your laptop always has its own MAC address
- Your laptop's IPv6 address will always relate to its own MAC address
- Therefore, your laptop can be used to track your movements



Autoconfiguration for IPv6

- There are privacy considerations with SLAAC.
For example:
 - Your laptop always has its own MAC address
 - Your laptop's IPv6 address will always relate to its own MAC address
 - Therefore, your laptop can be used to track your movements
- For this reason, RFC4941 allows generating a set of temporary interface identifiers that would then be combined as before with the /64 subnet address to form IPv6 addresses



Autoconfiguration for IPv6

- There are privacy considerations with SLAAC.
For example:
 - Your laptop always has its own MAC address
 - Your laptop's IPv6 address will always relate to its own MAC address
 - Therefore, your laptop can be used to track your movements
- For this reason, RFC4941 allows generating a set of temporary interface identifiers that would then be combined as before with the /64 subnet address to form IPv6 addresses

● Autoconfiguration for IPv6—DHCP for IPv6

- DHCPv6 is the stateful address autoconfiguration protocol.
- The DHCPv6 client creates an identity-association (IA) consisting of a set of related IPv6 addresses and assigns it an IA identifier (IAID)
 - Each network interface that the DHCPv6 client will request an IPv6 address for must have at least one IA associated with it
- The client sends a Solicit message to locate any available DHCPv6 servers, the Solicit message includes the IA for each interface the client wants information for
- Any server that can meet the client's request sends an Advertise message to the client that includes IP address assignments plus additional resources
- The client chooses one of the servers and sends it a Request message, to request the configuration information (with IP address) from the server
- The server sends back a Reply message that includes the assigned IP addresses and the configuration information
- Note that DHCPv6 may or may not include IP addresses, the IP addresses could be assigned in a different way, perhaps using SLAAC or perhaps statically assigned



Virtual Local Area Network (VLAN)

- A Virtual Local Area Network (VLAN) is a set of devices that are logically isolated such that they act as if they are on a single LAN, even though they may be (somewhat) geographically distributed
- This happens at protocol layer 2
- A broadcast domain is a set of devices that can broadcast to each other at layer 2
 - A layer 3 router does not forward a broadcast frame, so it forms a boundary on a broadcast domain
- A virtual local area network (VLAN) is a group of hosts that communicate as if they were attached to the same broadcast domain regardless of their physical location



Virtual Local Area Network (VLAN)

- On a switch that handles VLANs, ports are associated with a VLAN number.
 - The switch only allows data to be sent between ports that are on the same VLAN
- To have a VLAN connected between two switches, a VLAN tag is used in the layer 2 frame header
 - For an outgoing packet, a switch adds the VLAN tag to the layer 2 frame header.
 - The receiving switch reads the VLAN tag and sends the data to the appropriate port
- A device on a VLAN is not able to connect to a device that is not on that VLAN without going through a level 3 router



Virtual Local Area Network (VLAN)

- Advantages of a VLAN:

- reduces the size of the broadcast domain,
 - results in less wasted bandwidth
 - frames don't get sent to (as many) computers that aren't interested in seeing them
- increases security
 - the traffic on a VLAN is not visible to devices not connected to the VLAN
 - Could have a “company” VLAN vs. a “guest” VLAN



Universally Unique Identifiers (UUIDs)

- First created in the 1980s by a graphical workstation manufacturer (Apollo Computers) as part of a system called the Network Computing System (based on the Network Computing Architecture)
- Later became part of the Open Software Foundation's Distributed Computing Environment
- Later defined in an Internet Engineering Task Force Standard



Universally Unique Identifiers (UUIDs)

Some uses of UUIDs :

- record identifiers in databases
- in Linux file systems
- To identify storage devices on Solaris



Universally Unique Identifier (UUID)

- A UUID is a 128 bit number (16 bytes/octets)
 - An example would be:
 - abcd1234-fae3-12cd-a4be-1234abcd4321
 - 8 digits, 4 digits, 4 digits, 4 digits, 12 digits
 - Lower case hexadecimal
 - The digit in red in the 3rd group shows the version # (1, 2, 3, 4, or 5)
 - The first two bits of the digit in red in the 4th group show the variant. “10” is the variant from RFC4122, so this digit would be 8, 9, a, or b



Universally Unique Identifier (UUID)

○ Variants from RFC 4122:

MSB	MSB -1	MSB -2	Description
0	X	X	Network Computing System (NCS) backward compatibility
1	0	X	RFC 4122 Variant
1	1	0	Microsoft backward compatibility
1	1	1	Reserved for the future



Universally Unique Identifier (UUID)-cont'd

- There are 5 versions of UUIDs:
 1. Time-based+MAC address
 2. DCE
 3. Name-based with MD5 hash
 4. Random (what Keystone uses)
 5. Name-based with SHA-1 hash



MD5 Message Digest Hash and SHA-1 Hash

- Even small changes in the message will (usually) result in a mostly different hash
- MD-5 Hash
 - 32 digit hexadecimal number
- SHA-1 Hash
 - 40 digit hexadecimal number
- SHA-1 considered safer than MD-5



Version 1: Time-Based UUID

- Timestamp is 60 bit unsigned integer (15 hex digits), Coordinated Universal Time, representing 100 nanosecond intervals since 15 Oct 1582 (date of Gregorian calendar reform).
- Clock ID is a 14 bit unsigned integer, initially (once in a system lifetime) initialized to a random number.
- MAC address is 48 bit unsigned integer, in 6 sets of two hexadecimal digits

Version 1: Time-Based UUID (cont'd)

- Given: abcd1234-fae3-12cd-a4be-1234abcd4321
- Time is: 0x2cdfae3abcd1234
 - Time_hi = 0x2cd, Time_mi=0xfae3, Time_low=0xabcd1234
- Clock ID is: 10 0100 1011 1110 base 2
 - Clock hi is 10 0100 (MSB is version 10, gives a4) Clock lo is 0xbe
- MAC address is: 12:34:ab:cd:43:21

Byte #	0-3	4-5	6-7	8	9	10-15
	Time_low	Time_mid	Time_hi & version	Clock hi & reserved	Clock low	Node (MAC) address
	abcd1234	fae3	12cd	a4	be	1234abcd4321



Version 4: Random UUID

XXXXXXXX-XXXX-**4**XXX-**a**XXX-XXXXXXXXXXXX

- Where the red 4 indicates the version number (as before)
- The red a indicates the variant (as before, the first two bits are “10” which means this value can be 8, 9, a, or b)
- The other hex digits, represented by x, are randomly generated.