CS 250: FOUNDATIONS OF COMPUTER SYSTEMS [NETWORKING]

Private Addresses
Need an address
for this *and* that?

Do more with less
Keep it private and use a NAT

For you don't need permissions
Nor worry about collisions

SHRIDEEP PALLICKARA
Computer Science
Colorado State University

COMPUTER SCIENCE DEPARTMENT   COLORADO STATE UNIVERSITY

1

Topics covered in this lecture

- Private IP Addresses
- DNS
- OSI Network architecture

COLORADO STATE UNIVERSITY   Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT   NETWORKING   L23.2

2

PRIVATE IP ADDRESSES

3

Issues with having a limited number of IPv4 addresses

- The number of available IP addresses is **limited**
  - Most internet service providers (ISPs) only assign a single IP address to a customer
- This IP address is assigned to the device that's directly attached to the ISP's network, usually a router
- However, many customers have multiple devices on their home network

4

We will see how multiple devices can share a single public IP address

- Private IP addresses
- Network Address Translation
- Note:
  - Most routers sold to consumers for home use are NAT routers, often with built-in wireless access point capabilities as well

Private IP addresses

- Certain ranges of IP addresses are considered **private IP addresses**
  - Addresses intended to be used on private networks; e.g., homes or offices
  - The devices aren't directly connected to the internet
- Any address that matches the pattern of **10.**x.x.x, **172.16.**x.x, or **192.168.**x.x is a private IP address
  - Anyone can use these ranges of IP addresses without asking permission
- The catch is that private IP addresses are **nonroutable**
  - Can't be used on the public internet
- For IPv6? Addresses starting with **fc**

Who assigns these addresses

- Private IP addresses are intended to be used simultaneously on multiple private networks
  - Unlike public IP addresses that must be unique
- It doesn't matter if multiple networks use the same addresses
  - The addresses **won't ever be seen outside** of the private network anyway

Who assigns these addresses

- A DHCP server on a home network can assign these addresses
  - Without worrying about whether any other network is using the same addresses
- Private IP addresses solve the problem of an ISP only providing a single public IP address to a home or business
- But how are private IP addresses useful if they aren't routable on the internet?

Network Address Translation (NAT)

- NAT allows devices on a private network (e.g., a home network) to all use the same public IP address on the internet
- As packets flow through a NAT router, the router modifies the IP address information in those packets

Comings and goings

- When a packet originating from the private home network arrives at the NAT router?
  - The NAT modifies the source IP address field to match the public IP address
- When a response comes back to the router?
  - The NAT sets the destination IP address to the private address of the host that originated the request

The result of these IP modifications?

- All traffic from the home appears to originate from the same public IP address
  - Even if there are actually multiple devices on the private network

NAT also has the side benefit of security

- The devices on the private network aren't directly exposed to the public internet
  - A malicious user on the internet can't initiate a connection directly to a private device

Private IP addresses are valuable not only for home networks

- Businesses that don't want their computers exposed to the public internet use them as well
- Many corporate networks use a **proxy server** rather than a NAT router

Proxy servers

- A proxy server is similar to a NAT router in that it allows devices on a private network to access the internet
- But a proxy server differs in that it typically operates at the application layer rather than the internet layer
- Proxies also usually provide additional features such as user authentication, traffic logging, and content filtering

DNS

COMPUTER SCIENCE DEPARTMENT COLORADO STATE UNIVERSITY

15

Why DNS?

- We've seen that hosts on the internet are identified by IP addresses
- However, most users of the internet rarely, if ever, directly deal with IP addresses
  - IP addresses work well for computers, not so much for humans
  - No one wants to remember sets of four numbers (six for IPv6) separated by periods
- DNS to the rescue!

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING L23.16

16

Domain Name System (DNS)

- DNS is an internet service that **maps** names to IP addresses
- This allows us to refer to a host by a name like www.example.com rather than by its IP address
- A computer's full DNS name is known as a fully qualified domain name (FQDN)

Domain Name System (DNS)

- travel.example.com is an FQDN
  - The name is composed of
    - A short, local hostname (travel)
    - A domain suffix (example.com)
- The term hostname is often used interchangeably to mean either the computer's short name or the FQDN

Difference between domains and hosts on that domain

- A domain, like example.com, represents a grouping of network resources managed by an organization
- Both example.com and travel.example.com are domain names
  - The former represents a network domain
  - The latter represents a specific host on that domain

Resolving hostnames

- Software needs to be able to query DNS to convert hostnames to IP addresses
  - Resolving a hostname
- To enable this functionality, hosts are configured with a list of the IP addresses of DNS servers
- This list is usually provided by **DHCP** (Dynamic Host Configuration Protocol)
  - Typically, is composed of DNS servers maintained by the internet service provider
  - Or running on the local network

Connecting to a server

- When a client wants to connect to a server by name, it asks a DNS server for the IP address corresponding to that name
  - The server replies with the requested IP address, if it can
  - Once the client has the server's IP, it proceeds to communicate with the server using the IP address

There need not be a one-to-one mapping between IP addresses and names

- A name can **map to multiple IP addresses**
  - Here, different clients query DNS for a certain name, and they may all receive a different IP address as a response
- Useful for situations where the service load needs to be distributed across multiple servers
  - Can be done geographically
    - Clients in Europe, for e.g., get a different IP address than clients in US and Asia
  - Allowing clients in each region to connect to the IP address of a server that's **physically close** to them

Multiple names can map to the same IP address

- A query for different names may return a single IP address
- This is useful when a server hosts multiple instances of the same type of service, each identified by name
- Common in **web hosting**, where a single server hosts multiple websites, each identified by its DNS name

Each entry in DNS is known as a record

- There are various kinds of records
- The most basic is an **A record**, which simply maps a hostname to an IP address
- **CNAME** (canonical name) records map one hostname to another hostname
- **MX** (mail exchanger) records used for email services

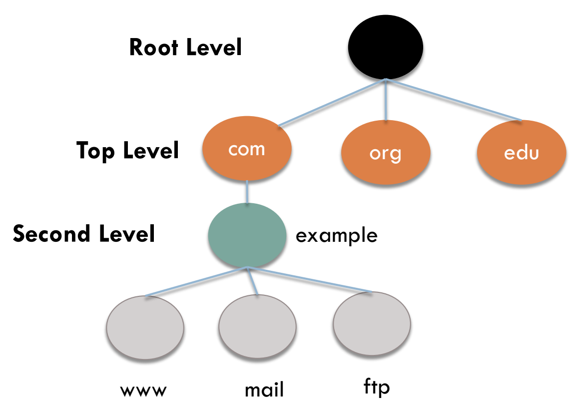
Getting DNS to scale

- No single organization would want to undertake the task of managing the many, many DNS records that exist today
  - DNS is implemented in a way that allows for **shared responsibility**
  - A DNS name like www.example.com actually represents a **hierarchy** of records
- Different DNS servers are responsible for maintaining the records at different levels of the hierarchy

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING L23.25

25

An example



COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING L23.26

26

DNS Hierarchy

- At the top of this hierarchical tree is the **root domain**
- The root domain doesn't get a textual representation in a DNS name like www.example.com
  - But it's an essential part of the DNS hierarchy
- The root domain contains records for all the **top-level domains** (TLDs) like .com, .org, .edu, .net, and so forth
- There are **13 root name servers** worldwide, each responsible for knowing the details of all the top-level domain servers

Resolving a name [1/2]

- Let's say you want to look up a record in a domain that ends with .com
- A root server can point you to a TLD server that knows about domains under .com
- A top-level DNS server is responsible for knowing about all the second-level domains under its hierarchy
  - A top-level DNS server for .com could point you to the second-level DNS server for example.com

Resolving a name [2/2]

- The DNS servers for second-level domains maintain records for hosts and third-level domains that fall under second-level domains
  - This means that the DNS server(s) for example.com are responsible for maintaining the records for hosts like www.example.com and mail.example.com
- This pattern continues, allowing for **nested domains**
  - Once a domain is registered under a top-level domain, the owner of that domain can create as many records as needed under their domain

DNS and Caching [1/2]

- When a computer needs to find the IP address for an FQDN, it sends a request to its configured DNS server
- If the server has recently looked up the requested record?
  - May have a copy of that record stored in its **cache**
  - Can immediately return the IP address to the client

DNS and Caching [2/2]

- If the DNS server doesn't have the response in cache
  - It may query other DNS servers as needed to get the answer
    - Starting at the root
    - Working down the hierarchy of servers to find the record in question
- Once the server has the record, it can cache it so that it can immediately respond to future queries for that record
  - Eventually the cached record is removed
    - To ensure that the server always provides reasonably recent data

OSI NETWORK ARCHITECTURE

OSI network architecture

- Model is a product of the Open Systems Interconnection (OSI) project
  - At the International Organization for Standardization (ISO)
- Partitions network functionality into **7 layers**
- Physical Layer
  - Handles transmission of **raw bits**
  - Standardizes electrical, mechanical, and signaling interfaces
    - **0** bit should be received as **0** not **1**

OSI network architecture: Data link Layer

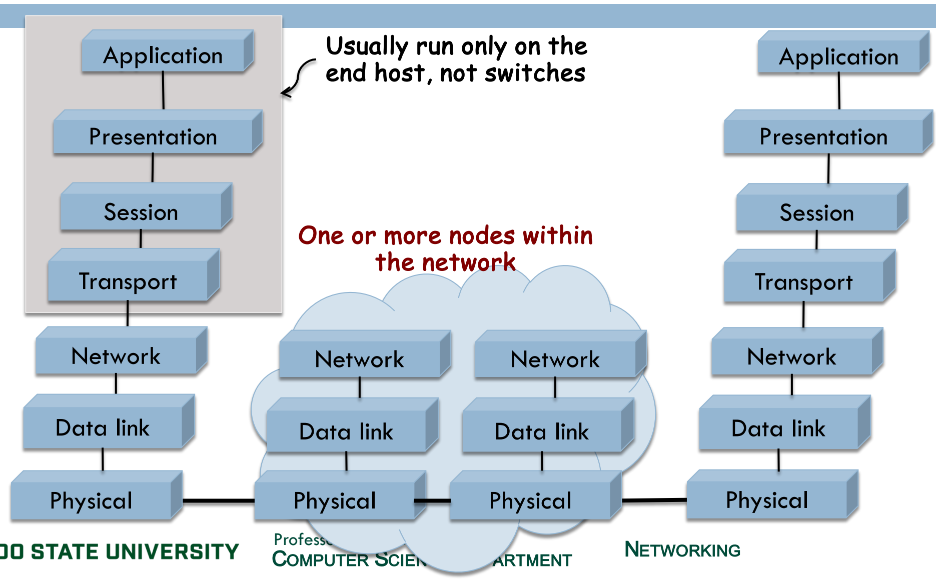
- Collects stream of bits into a **frame**
  - Puts special *bit pattern* at the start/end of each frame
  - Frames, not raw bits, are delivered to host
- Compute **checksum** for frame
  - Check for correctness and request retransmission
- Network adaptors & device drivers implement this

OSI network architecture

- Network layer
  - Handles routing among nodes in a **packet-switched** network
  - Unit of data exchanged is **packet** not frames
- Layers implemented on all network nodes?
  - Physical, data and network

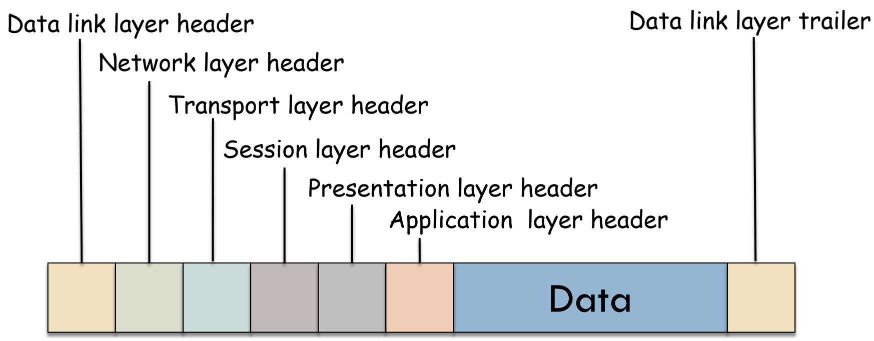
35

OSI Architecture



36

How messages flowing through the OSI stack will appear on the network



OSI network architecture

- Transport
  - Implements process-process **channel**
  - **Messages** {not packet or frame}
- Presentation
  - **Format** of data exchanged between peers
- Session
  - **Namespace** to tie different transport-streams that are part of the application

The contents of this slide-set are based on the following references

- Matthew Justice. *How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine*. ISBN-10/ISBN-13 : 1718500661/ 978-1718500662. No Starch Press. 2020. [Chapter 11]