09 - Naming

Overview

Protocols

- DHCP (Dynamic Host Configuration Protocol) - to get an IP addr
  - MAC → IP (given MAC, provision IP)
  - any node can request from multicast DHCP server → this means DHCP mac addr and IP begins with leading 1 (multicast)
- DNS (Domain Name Server) - map domain names to IPs
  - domain name ⟷ IP
  - given from DNS, hierarchical until root DNS
  - root DNS (.com, .uk, etc.)
  - Top Level Domain (TLD) e.g. google, apple, etc.
  - sub domains - r.g., docs.google.com
- NAT (Network Address Translation) - proxy a network (multiple users) through a single IP
  - Private IP ⟷ Public IP
  - simpelst is 1:1 mapping
  - stored on local/private router
- ARP (Address Resolution Protocol) - map IP to MAC addr
  - IP → MAC (given IP, reply with MAC)
  - every router has a ARP table for translations

Names

- Domain/Host name - user readable, variable length, hierarchical
- IP addr - 32 bit, topologically unique, hierarchical (subnetting)
- MAC addr - 48 bit, fixed name, globally unique

Translation Patterns

- DNS -
- ARP -
- DHCP -

DHCP

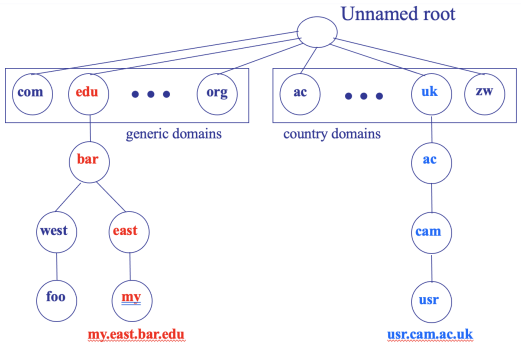
- DHCP (Dynamic Host Configuration Protocol) - to get an IP addr
  - DHCP request
    - Broadcast-based LAN protocol algorithm
      - Host broadcasts “DHCP discover” on LAN (e.g. Ethernet broadcast)
      - DHCP server responds with “DHCP offer” message
      - Host requests IP address: “DHCP request” message
      - DHCP server sends address: “DHCP ack” message w/IP address

DNS

DNS (Domain Name Server) - map domain names to IPs

- Distributed administrative control
  - Hierarchical name space divided into zones
  - Distributed over a collection of DNS servers
- Hierarchy of DNS servers
  - Root servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Performing the translations
  - Local DNS servers
  - Resolver software

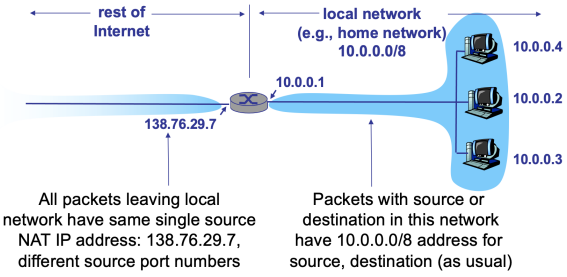
- overview
- radix tree structure for DNS hierarchy (almost always cached after 1t request to root)

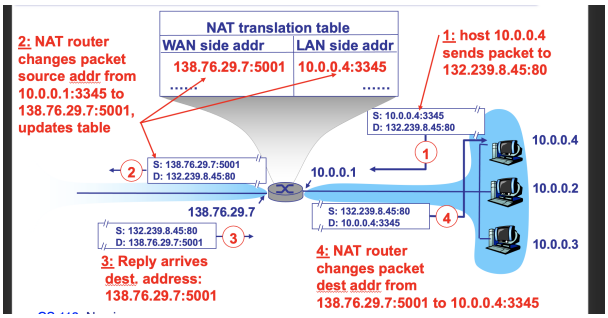


- DNS replicated to many servers, use UDP queries for robust requests, use exponential backoff for requests to the same server
- SD-DNS e.g., Akami server store shortest paths as SDN/CDN and propagate (similar struture to other SD-WAN)

NAT

- NAT (Network Address Translation) - proxy a network (multiple users) through a single IP
  - NApT - Network Address and Port translation
  - PAT - port address translation
  - NAT encompasses all
- due to IP limits and flexibility to change ISP, local networks manage a private IP space using bogons
- bogons - IPs with prefixes that indicate private IPs e.g., 192.168 ... , 176.1 ...
- local routers (closest to private network) store private to public IP mappings in a table
  - route packets based on mapping
- simplest case is 1-to-1 mapping, in some clever cases map set of private IPs to smaller set of public IPs
- in some cases map all private IPs to 1 public IP
- example NATed network



- NApTed example 

- NATs may be layer-violating as they may change existing protocols like FTP to support private address mappings (e.g., when IP in application data)

The trick to Many:1

- Mapping many private addresses to 1 public addr makes it impossible to anti-alias which actual device should receive the packet
- Hack is to use the TCP Port ⇒ 32 bit IP + 32 TCP ports = 64 bit address space
- enables port forwarding for private IP to be directly accessible over the internet
  - other forms of NAT traversal also possible