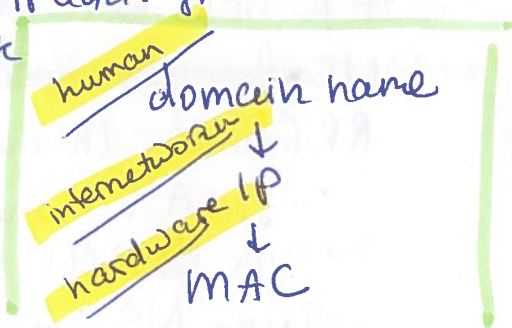
# lecture 8

application layer

## Protocols for addressing and name:

### DHCP (Dynamic Host Configuration Protocol):

- goal:
  - allow host to dynamically obtain its IP addr from network server when it joins network
  - renew its lease on addr in use
  - reuse of addr
  - support of mobile users




- overview:
  - host broadcasts "DHCP discover" msg
  - DHCP server responds with "DHCP offer" msg (optional)
  - host requests IP addr with "DHCP request" msg
  - DHCP server sends addr with "DHCP ack" msg

- returns:
  - addr of first-hop router for client (default router)
  - name & IP addr of DNS server
  - network mask (host partition of addr)

### ARP (Address Resolution Protocol)

- goal:
  - routing to another LAN / same LAN
  - determine interface's MAC addr knowing its IP addr → ARP Table

- ARP Table
  - each IP node (host/router) on LAN has table (may be integrated with forwarding table)
  - IP/MAC addr mappings for some LAN nodes: <IP/MAC/TTL>
  - TTL (Time to Live): time after which addr mapping will be forgotten (≈ 20 min)

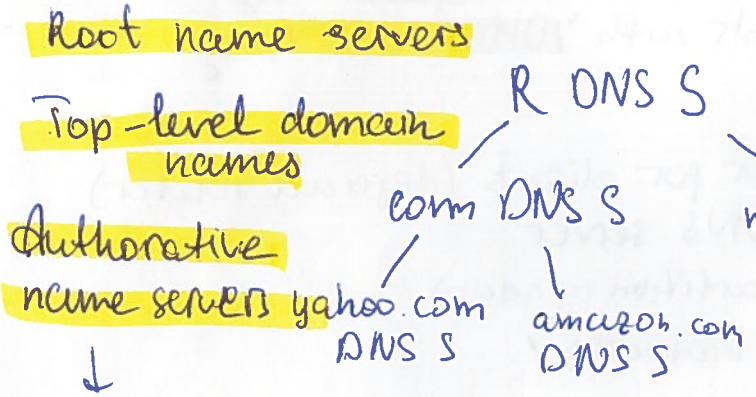
**same LAN**


B's MAC is not in A's ARP Table
- A broadcasts ARP Query with B's IP
- B replies to A with its MAC
- A caches IP-to-MAC in ARP Table in soft table (until refreshed or TTL)

**another LAN**


- IP datagram: src A, dest B
- A creates link-layer frame with R's MAC as dest and IP datagram
- R creates IP datagram: src A, dest B
- R creates link-layer frame with B's MAC as dest and IP datagram
- B extracts IP datagram from received frame

- DNS (Domain Name System)
  - hostname to IP address translation
  - host aliasing
  - mail server aliasing
  - load distribution: set of IP addresses for one canonical name
  - not possible to centralize it
- DNS Resource Records
  - RR format: (name, value, type, class, ttl)
    - Type A: name hostname, value IP addr
    - Type CNAME: name alias name, value canonical name
    - Type NS: name domain, value hostname of authoritative ser
    - Type MX: value is name of mailserver associated with nam
- Hierarchical Database




TLD servers cached in LNS (root not often visited)

  - multiple level hierarchy possible
  - hostname to IP mappings for organiz'l. servers (Web, mail)
- Local Name Server
  - does not necessarily belong to hierarchy
  - each ISP has one
  - when host makes DNS qu query is sent to its LNS a.k proxy to forward to hiera

Routing Protocols (Applying only one Protocol in Internet is not possible) ! hierarchy

- Link State Routing
  - flood local link-state information to all nodes
  - use local algorithm to calculate route to all destinations
- Flooding
  - each node creates link state packets (LSPs)
    - id of src node
    - cost to direct neighbor
    - sequence number
    - TTL
  - sequence number is incremented as node creates new LSP
    - before TTL expired
    - if link to neighbour disappered / cost increased
  - use of ACKs and retransmissions
  - for received LSP: store or update
  - for stored LSP: forward to neighbours except ~~receiver~~ sender
  - decrement TTL for stored / forwarded LSPs
  - if TTL == 0 => drop LSP

Route Calculation Algorithm (Forwar Search Dijkstra)

1. Confirmed list with cost to itself (0) and empty Tentative list
2. For node added to Conf. l (NEXT), select its LSP
3. For each Neighbour of Next calculate cost from myself to Next to Neighbour
   - 3a. if Neighbour is not in any list → add it to Tentative list (Neighbour, Cost, NextHop)
   - 3b. If Neighbour is on Tentative list ss Cost is less than in → replace entry (Neighbour, Cost, NextHop) confirmed
4. If Tentative list is empty → stop
   else pick entry from Tentative list with lowest ~~store~~ Cost and do step 2.

Intradomain Internet

- OSPF (Open Shortest Path First)
  - auth of all msg
  - additional hierarchy
  - load balancing