

Weekly Log

Week-8

Network Layer DHCP

- Beberapa komponen yang terdapat pada network layer yaitu: DHCP, NAT, ICMP, Traceroute, IPv6.
- Ada beberapa cara komputer/host dalam mendapatkan IP Address:
 - Static IP Address: Secara manual memasukkan IP Address, dan harus mengerti subnetting” yang diajarkan di module sebelumnya. Biasanya digunakan dalam IP yang tidak berubah-ubah/IP Public.
 - Dynamic IP Address: Secara automatic memasukkan IP Address, dynamic IP Address menggunakan teknik DHCP. Biasanya digunakan dalam IP Address yang berubah-ubah.
- Static IP Address biasanya dikelola oleh beberapa server-server karena IP Address mereka bersifat tetap dan tidak berubah-ubah.
- Dynamic IP Address (DHCP) biasanya digunakan oleh pengguna jasa internet (Client) karena IP address yang berubah-ubah.

1. DHCP (Dynamic Host Configuration)

Goal dari DHCP adalah memungkinkan host untuk mendapat IP address secara dynamic dari network server ketika host telah masuk ke dalam network.

DHCP Overview:

- host broadcasts “DHCP discover” msg [optional]
- DHCP server responds with “DHCP offer” msg [optional]
- host requests IP address: “DHCP request” msg
- DHCP server sends address: “DHCP ack” msg

Contoh DHCP:

1. Connecting laptop ke IP Address, lalu address tersebut di assign ke first-hop router dan address tersebut dari DNS Server.
2. DHCP request dienkapsulasi di dalam UDP (paket-paket di enkapsulasi). Lalu dienkapsulasi ke ethernet dan di broadcast ke semua jaringan
3. DHCP akan mengembalikan offering ke client tersebut dengan IP Address
4. Lalu Client mencatat IP tersebut dan melakukan request IP Address ke DHCP Server
5. Selanjutnya DHCP akan memberikan kembali IP Address tersebut ke client sehingga client dapat mengetahui IP Address , dll sehingga client dapat menulis network interfacenya.

2. NAT (Network Address Translation)

- NAT adalah sebuah metode untuk mentranslate IP Address yang diberikan oleh DHCP server agar client dapat berselancar di internet dan seakan-akan kita menggunakan IP public.

- Semua datagram yang meninggalkan local network dia akan menggunakan single source NAT IP address (IP public).
- Motivasi:
 1. Range address itu sangat jarang dan langka (1 Address untuk semua device).
 2. Bisa merubah address device di local network, mau sebanyak apapun kita bisa ubah (biasanya 1 router bisa 250 user).
 3. Kita dapat menggunakan ISP tanpa mengganti address dari local network kita karena semua sudah disetting
 4. Device dalam local network tidak dapat diakses dari luar sehingga butuh translasi agar bisa diakses dari luar
- Implementasi NAT router harus:
 1. Outgoing datagram: replace
 2. *Remember (in NAT translation table)*
 3. Incoming datagram: replace

3. ICMP (Internet Control Message Protocol)

- ICMP (Internet Control Message Protocol) digunakan sebagai suatu protokol yang digunakan agar kita bisa mengontrol atau mengetahui status dari message kita. ICMP biasanya digunakan untuk host atau router.
- Traceroute digunakan mengetahui status dari tempat kita ke sebuah server (hopping berapa kali dihitung dalam TTL, biasanya 3 kali). Ketika ICMP message sampai maka source akan merecord RTTs. Stopping criteria terjadi jika UDP segment telah sampai ke destination host, kita bisa membatasi jumlah hop-nya atau "port unreachable". Dan terakhir source stop.

4. IPV6

- IPv6 merupakan salah satu solusi yang bisa menjembatani ketika ketidakmampuan 32bit IPv4. IPv6 mempunyai fixed length dengan besarnya 40byte header dan fragmentasi tidak diperbolehkan.
- Datagram format pada IPv6
 1. *priority*: identify priority among datagrams in flow
 2. *flow Label*: identify datagrams in same "flow."
 3. *next header*: identify upper layer protocol for data
- IPv6 memiliki source address dan destination address sebesar 128 bits
- Beberapa hal yang diubah dari IPv4 ke IPv6:
 1. *checksum*: removed entirely to reduce processing time at each hop
 2. *options*: allowed, but outside of header, indicated by "Next Header" field
 3. *ICMPv6*: new version of ICMP
 - additional message types, e.g. "Packet Too Big"
 - multicast group management functions