

RANGKUMAN PRAKTIS

Jaringan Komputer

Topics

- End-to-end Protocol
- Congestion Control
- End-to-end Data
- Network Security

Bab 4: Advanced Internetworking (sekilas aja)

Global Internet

Selama ini yang dipelajari tuh baru sampai routing biasa, yang setiap router-nya harus tau semua network yang terkoneksi ke internet (which agak mustahil). Nah makanya dibikin suatu hierarki jaringan.

Hierarki-nya dulu cuma dibagi per daerah, nama-nya regional networks. Regional network ini terkoneksi lewat suatu backbone (suatu area network yang jadi perantara buat area-area lainnya, as the name suggests, dia tulang punggung-nya) nationwide.

Setiap provider dan end user itu biasanya entitas independen, jadinya beda provider, bisa aja beda protokol routing dari dalam jaringan, dll. Makanya dinamain Autonomous System (AS), basically sebuah entitas independen dengan yang lain.

Dibahas tentang scaling, yang harus consider dua hal, scalability dari routing (cari cara buat minimalisir jumlah jaringan yang perlu disimpan di routing table) dan address utilization (jangan terlalu cepat habis address space dari IP)

Bab 5: End-to-end Protocol

Tentang transport protocol, pokoknya gimana caranya biar network yang unreliable itu bisa jadi reliable di layer atasnya ini.

UDP

Yang paling simpel protokolnya itu UDP, buat komunikasi antar proses (pake PID dari OS) terus dikirimnya ke host, port pair, jadi bisa lebih mudah demultiplexingnya. Cara tau portnya? Pakai port yang umum untuk hal tersebut.

Di UDP, gaada flow control mechanism, dan dia delivery-nya ga reliable, tapi dia mastiin bener pake checksum.

TCP

TCP itu lebih reliable, harusnya gaada missing ato out of order data (retransmisi dan nunggu data yang dilakuin di sliding window). TCP bisa multiplexing, dan dia masing-masing arah ngirim punya byte stream sendiri. Dia ada flow control, dan juga congestion control. Flow control itu ngontrol flow data-nya yang diterima oleh penerima, kalau congestion control itu ngontrol traffic network-nya.

Nah TCP tuh punya kaya link logical, tapi karena link nya logical, makanya ada handshake dan ada termination.

Segment Format

Ada format byte yang dikirim sama TCP, ini namanya segment

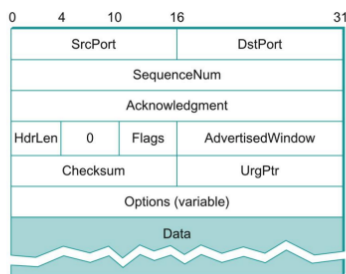


FIGURE 5.4 TCP header format.

Connection Establishment and Termination

Three way handshake

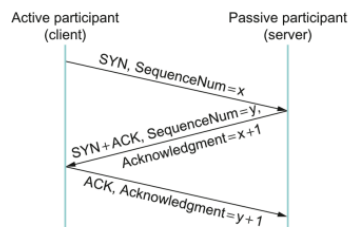


FIGURE 5.6 Timeline for three-way handshake algorithm.

Sliding Window

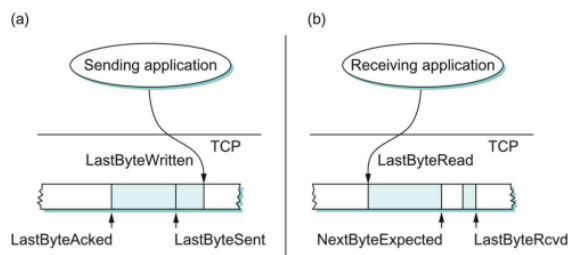


FIGURE 5.8 Relationship between TCP send buffer (a) and receive buffer (b).

Flow Control

Intinya ini pake effective window, ada receive buffer, ada advertisement buffer (yang ngirim)

Receive buffer \geq byte received - byte read, jadi kalau dia belum baca lagi, gaboleh nerima lagi dulu

Advertisement buffer \geq byte sent - byte acked, jadi kalau belum di ack, gaboleh ngirim lagi dulu

Sequence Number Wraparound

Biar ga terjadi, bisa ada beberapa mekanisme

1. Window Scaling
Memperluas jangkauan seqnum effective, jadi nambahin seqnum sebelum harus reset
2. Gunakan sequence number besar
3. Maximum Segment Lifetime (MSL)
Waktu maksimum paket TCP diizinkan buat beredar sebelum dibuang. Mencegah paket lama dari koneksi sebelumnya dari muncul di koneksi baru (2 menit biasanya)
4. Time-wait state
Disuruh nunggu $2 \times \text{MSL}$ setelah koneksi ditutup, biar kalau paket duplikat di jaringan ilang dulu, jadi kaya nunggu semua mati dulu
5. Ditangani langsung
Langsung reset dari 0 lagi, dengan pengurutan relatif, bukan absolut

Ringkasnya

TCP

End-to-end Issue

Masalah yang dihadapi dalam menyediakan komunikasi antara dua endpoint di Internet, kaya out of order, missing, duplikasi, dan retransmisi

... skip sampai setelah sliding window

Triggering Transmission

Silly window syndrome: masalah yang terjadi kalau data dikirimnya dengan potongan yang sangat kecil, yang mengakibatkan overhead tinggi dan efisiensi rendah

Nagle's Algorithm: Algoritma buat ngurangin masalah silly window, dengan menunda pengiriman data sampai ada cukup data di buffer, atau sampai semua data yang dikirim sebelumnya sudah di ack.

Adaptive Retransmission

Original algo: Cara pertama yang TCP pake buat nentuin kapan paket yang hilang harus dikirim ulang

Karn/Partridge Algorithm: Perbaikan algo yang tidak mengandalkan pengukuran RTT dari paket yang diulang untuk pengaturan timer

Jacobson/Karels Algorithm: Algoritma yang lebih optimal lagi, pengoptimalan pengaturan timer berdasarkan variabilitas dalam estimasi RTT

RPC

RPC Fundamental

Identifier: Gimana cara tau prosedur mana yang mau dipanggil? Pakai identifier, yaitu nama prosedur atau nomor unik “procedure number” untuk mapping request dan procedure.

Network Limitations: RPC kan abstraksi, nah buat mengatasi keterbatasan latensi atau low reliability, RPC pakai teknik kaya timeout, pengulangan, sama batch processing

Synchronous vs Asynchronous: sesuai namanya, rpc bisa sinkron dan asinkron

RPC Implementation

SunRPC: Open Network Computing RPC, jadi dasar buat NFS

DCE-RPC: Distributing Computing Environment, mendukung keamanan, skalabilitas, dan integrasi

gRPC: dari google, RPC framework modern yang mendukung komunikasi microservices, basisnya HTTP/2, pakai protobuf untuk encode data

Real-Time Protocol

Requirements

Keperluannya itu buat real-time transfer data, kaya audio dan video. Jadi harus ada dukungan buat ngirim data dengan jitter rendah, nanganin paket yang hilang tanpa retransmisi, dan penyesuaian dengan tingkat bandwidth

RTP Design

Header Format mengantung beberapa field penting seperti seqnum, timestamp, payload type, identifier sinkronisasi sumber yang membedakan antar sumber dalam sesi yang sama

Control Protocol

RTCP bekerjasama dengan RTP untuk menyediakan kontrol dan pemantauan jaringan. Jadi RTCP itu bakal ngumpulin statistik transmisi data, kaya hilang paket dan jitter, dan ngasih feedback buat kualitas layanan.

Bab 6: Congestion Control

Ringkas aj yh

Issues in Resource Allocation

Network Model

Packet-switched network: model jaringan dimana data dikirim dalam bentuk paket yang independen, jadi sumber daya jaringan bisa dibagi secara dinamis

Connectionless Flows: aliran daya tidak memerlukan koneksi tetap sebelum data dikirim, mirip UDP

Service Models: model layanan berbeda yang ditawarkan oleh jaringan, seperti Quality of Service, yang mempengaruhi bagaimana sumber daya dialokasikan dan dikelola.

Taxonomy

Router centric vs Host centric: congestion control yang dilakukan oleh router dan yang dilakukan oleh host.

- **Queue Management:** Router mengatur antrian paket dan memutuskan paket mana yang harus diantre atau dibuang.
- **Active Queue Management (AQM):** Router menggunakan algoritma seperti Random Early Detection (RED) untuk mendeteksi kemacetan sebelum antrian menjadi penuh dan mulai membuang paket secara selektif atau memberi tanda pada paket untuk menyampaikan peringatan kemacetan kepada pengirim.
- **Explicit Congestion Notification (ECN):** Beberapa router mendukung ECN, di mana mereka tidak membuang paket tetapi sebaliknya menandai paket yang menunjukkan bahwa terjadi kemacetan.

Keuntungan dari pendekatan **router-centric** adalah bahwa router dapat **mengambil keputusan berdasarkan kondisi jaringan secara keseluruhan**, yang mungkin tidak terlihat oleh host. Namun, ini juga berarti router harus cukup canggih untuk melakukan tugas-tugas ini, yang bisa meningkatkan kompleksitas dan biaya.

- **Window Adjustment:** Algoritma kendali kemacetan seperti TCP Congestion Avoidance Algorithm mengatur ukuran window pengiriman berdasarkan pengakuan dari penerima dan penandaan kemacetan yang diterima.
- **Rate Limiting:** Host menyesuaikan laju pengiriman data berdasarkan umpan balik dari jaringan, yang mungkin termasuk pengakuan dari penerima atau pesan kemacetan dari router.
- **Retransmission Strategies:** Host menentukan kapan dan bagaimana mengirim ulang paket yang hilang, sering kali dengan mempertimbangkan estimasi waktu round-trip dan kehilangan paket.

Keuntungan dari pendekatan **host-centric** adalah bahwa host dapat **lebih responsif terhadap umpan balik jaringan** yang mereka terima secara langsung, dan pendekatan ini bisa lebih mudah diimplementasikan karena tidak memerlukan kemampuan lanjutan pada router. Namun, ini juga berarti bahwa host harus memiliki logika yang cukup untuk dapat membuat keputusan yang baik tentang bagaimana dan kapan mengirim data.

Reservation based vs Feedback based: reserve dulu resource nya vs nyesuain sumber daya berdasarkan kondisi jaringan saat ini

Window based vs Rate based: mengontrol jumlah paket yang dikirim dalam satu waktu vs mengontrol laju data yang dikirim

Evaluation Criteria

Effective Resource Allocation: kriteria untuk evaluasi efektivitas mekanisme alokasi sumber daya dalam mengoptimalkan penggunaan sumber daya dan memaksimalkan throughput jaringan

Fair Resource Allocation: seluruh aliran data mendapat bagian adil dari sumber daya jaringan tanpa merugikan yang lain

Additive Increase, Multiplicative Decrease

Kalau berhasil, nambah 1 window size nya, kalau gagal, di cut in half window size nya
Congestion Window

Slow start itu exponential increase

Fast retransmit, lebih cepet retransmit (3 duplicate ack)

Fast recovery, kalau ada paket yang hilang, kemungkinan macet. Jadi dirancang buat ngurangin window congestion, tapi tetep memungkinkan data terus lanjut tapi dengan laju yang dikurangi.

--	--

Soal Latihan

1. Sebutkan & jelaskan Elemen utama protokol layer Transport
 - a. Segmentasi dan Reassembling
Segmentasi itu memecah data menjadi segmen kecil biar gampang ngirimnya. Terus saat diterima dia di-reassemble jadi data awal
 - b. Control Flow
Mengatur laju pengiriman data, biar penerima ga kelebihan beban. Jadi si pengirim, ngirim-nya sesuai sama yang si penerima bisa proses
 - c. Error Handling
Harus bisa mendeteksi kesalahan yang terjadi selama transmisi data, sama mastiin data yang dikirim itu data yang bebas dari kesalahan (checksum, retransmission)

- d. Multiplexing dan Demultiplexing

Multiplexing itu proses gabungin data dari beberapa aplikasi jadi satu stream (di sisi pengirim), jadi 1 koneksi bisa untuk beberapa aplikasi

Demultiplexing itu proses misahin data, dari satu stream, diterima oleh beberapa aplikasi (di sisi penerima)
 - e. Connection Establishment and Termination

Harus bisa buka dan tutup koneksi dengan handal (three-way handshake)
 - f. Reliability and Ordered Delivery

Harus menjamin bahwa paket itu sampai di tujuan dengan benar, bebas dari kesalahan, dan sesuai urutan. Jadi harus di re-check dan dikirim ulang kalau perlu.
 - g. Congestion Control

Punya mekanisme buat mengontrol kepadatan jaringan, traffic yang dapat terjadi kalau data dikirim-nya terlalu cepat
2. Diasumsikan suatu protokol TCP yang menggunakan sliding window, namun memperbolehkan ukuran window melebihi 64 KB. Andaikan terdapat 1 file berukuran 8 MB akan ditransfer via TCP tersebut, dengan ukuran window penerima sebesar 1 MB. Jika TCP mengirimkan file tersebut per paket dengan ukuran paket sebesar 2-KB serta menggunakan mekanisme slow start, maka :
- a. Berapakah jumlah RTT yang terjadi dari awal transmisi hingga slow start berakhir (memperbolehkan ukuran window pengirim sebesar 1 MB) ?
 - b. Berapakah jumlah total RTT yang terjadi hingga proses transmisi file selesai?
- a. Untuk menjawab, perlu dipahami bagaimana mekanisme slow start dalam TCP, dimana saat mulai, ukuran window congestion TCP (cwnd) dimulai dari 1 segmen, meningkat secara eksponensial setiap RTT sampai mencapai threshold (ukuran window penerima)
- Setiap RTT, ukuran window akan dikali 2, yang perlu dihitung itu berapa RTT yang dibutuhkan sampai 1MB
- RTT 1: 2 KB \rightarrow 4 KB (cwnd dikalikan 2)
- RTT 2: 4 KB \rightarrow 8 KB
- RTT 3: 8 KB \rightarrow 16 KB
- RTT 4: 16 KB \rightarrow 32 KB
- RTT 5: 32 KB \rightarrow 64 KB
- RTT 6: 64 KB \rightarrow 128 KB
- RTT 7: 128 KB \rightarrow 256 KB
- RTT 8: 256 KB \rightarrow 512 KB
- RTT 9: 512 KB \rightarrow 1024 KB
- Jadi slow start selesai setelah 9 RTT
- b. Jumlah total RTT
- Ukuran file: 8 MB (8192 KB)

Ukuran Window Pengirim Max: 1 MB (1024 KB)

Paket per window max: $1024 \text{ KB} / 2 \text{ KB/paket} = 512 \text{ paket}$

Setelah slow start, TCP masuk ke fase congestion avoidance, dimana ukuran window meningkat secara linear setiap RTT. Tapi kan window pengirim udah sebesar window penerima (1 MB), jadi gabakal nambah lagi window size nya.

Total paket: $8192 \text{ KB} / 2 \text{ KB/paket} = 4096 \text{ paket}$

Paket per window: 512 paket

Jumlah window yang diperlukan: $4096 \text{ paket} / 512 \text{ paket/window} = 8 \text{ window}$

Kan setiap window perlu 1 RTT untuk konfirmasi penerimaan, dan udah 9 RTT buat mencapai window size maksimum, jumlah total RTT itu $9 + 8 = 17 \text{ RTT}$

$8192 - 2 \cdot 512 \text{ KB} = 8192 - 1022 = 7170$

$7170 / 2 = 3585 \text{ paket}$

$3585 / 512 = 7 \text{ window}$

RTT total = $9 + 7 = 16$

$$S_n = \frac{a(r^n - 1)}{r - 1}$$

3. Diketahui sebuah koneksi sliding window TCP memiliki ukuran window sebesar 4 segmen dengan RTT sebesar 200 ms.

Pengirim mengirimkan segmen dalam laju konstan yaitu 1 segmen per 100 ms, serta penerima mengirimkan balik ACK dalam laju yang sama tanpa delay.

Misalkan sebuah segmen hilang (segmen loss) dan dideteksi oleh algoritma fast retransmit pada saat penerimaan ACK duplikat ke-3.

Jika saat ini proses transmisi sedang berjalan pada saat ketika ACK dari segmen yang diretransmisi tiba di pengirim, berapakah selisih waktu antara transmisi dengan adanya 1 segmen hilang dibandingkan dengan transmisi tanpa kehilangan segmen (lossless transmission) dengan skenario: pengirim harus menunggu ACK dari retransmisi segmen yang hilang tersebut sebelum melanjutkan proses sliding window ? Jelaskan jawaban Anda.

Jawaban:

Pahami bagaimana sliding window TCP beroperasi, khususnya buat skenario loss segment terus pake algoritma fast retransmit

Skenario Normal (Lossless Transmission)

Ukuran window: 4 segmen

RTT: 200 ms

Laju pengiriman: 1 segmen per 100 ms

Karena menggunakan sliding window, pengirim akan menunggu ACK dari segmen pertama sebelum mengirimkan segmen kelima. Setiap segmen butuh 1 RTT untuk terima ACK. Dalam satu RTT, pengirim dapat ngirim 4 segmen.

Skenario Loss

Misal segmen pertama hilang, terus terdeteksi oleh penerima pada penerimaan ACK duplikat ke-3, berarti ACK duplikat untuk segmen pertama diterima setelah pengiriman segmen ke-4. Jadi begitu dapat ACK duplikat ke-3 (segmen 1 hilang tapi udah terima segmen 2 3 4), pengirim langsung fast retransmit.

Waktu ngirim 4 segmen pertama: $4 \text{ segmen} \times 100 \text{ ms/segmen} = 400\text{ms}$

Waktu terima 3 ACK duplikat: $3 \text{ segmen} \times 100 \text{ ms/segmen} = 300\text{ms}$

Waktu untuk ACK dari segmen yang diretransmisi tiba = 200ms (1 RTT, $2 \times 100\text{ms}$)

Total = $400 + 300 + 200 = 900\text{ms}$

Kalau lossless, 900ms tu bisa ngirim 9 segmen, tapi karena ada loss, jadi cuma bisa ngirim 4 segmen, terus retransmisi dan tunggu ack dari segmen yang diretransmisi.

Lossless 400ms untuk 4 segmen, Loss 900ms untuk 4 segmen (1 retransmisi)

ACK duplikat tuh terjadi karena penerima akan ngirim ACK sesuai dengan paket tertinggi yang berurutan.

4. Sebuah protokol sederhana berbasis UDP untuk melakukan download file berjalan sebagai berikut:
 - a. Client mengirim request untuk meminta sebuah file (memberikan nama file)
 - b. Server menjawab dengan mengirimkan paket data pertama
 - c. Client mengirimkan ACK, dan transfer berikutnya dilakukan dengan stop-and-wait, hingga seluruh file terkirim.

Kembangkanlah protokol di atas sehingga mampu menjamin autentikasi pengirim dan integritas pesan/data, yang kebal terhadap replay attack

Autentikasi dan Integritas dengan Penggunaan Kriptografi

- Digital signature dan certificate, buat autentikasi pengirim. Jadi sebelum transaksi, client dan server harus saling bertukar sertifikat publik.
- Hashing buat pastiin integritas data, jadi nanti hash-nya juga ikut dikirim, terus sama penerima di-hash ulang datanya lalu dibandingin dengan hash yang terkirim.

Pencegahan Replay Attack

- Menyertakan timestamp unik buat setiap pesan. Kalau timestamp-nya udah lewat, ga

bakal diterima.

- Nonce (Number used once), menggunakan nomor yang digunakan hanya sekali, yang unik untuk setiap sesi komunikasi. Buat mastiin setiap pesan dalam sesi itu unik dan gaboleh digunain lagi dalam sesi atau sesi lain.
- Session ID yang unik dalam setiap pesan, buat memastikan bahwa pesan berasal dari sesi yang sedang aktif.
- Sequence Number, nomor urut, kalau dia ga sesuai, ditolak

Secure Handshake

- Handshake protocol sebelum transaksi file
- Menggunakan metode key exchange yang aman

Tapi mainly, yang bisa difokusin kalau stop-and-wait itu, urutan paket-nya sih yang harus dipastiin urut. Soalnya dia punya waktu untuk “mikir” gitu, jadi kalau out of order bakal di-reject.

5. Jelaskan latar belakang mengapa muncul teknik CIDR (Classless Inter Domain Routing) yang digunakan untuk distribusi alamat IP dimana sebelumnya alamat IP telah dibagi-bagi menjadi kelas-kelas tertentu.

CIDR muncul karena kekurangan alokasi IP address intinya.

Dulunya kan pakai sistem kelas, tapi sistem kelas itu malah bikin boros IP address-nya, soalnya bisa aja butuh sejumlah IP diatas 254 (kelas C), tapi jauh dibawah 65.536 (kelas B)

Apalagi di IPv4 itu ada kekhawatiran kekurangan IP address, makanya dipikirin cara menggunakan alamat yang lebih hemat dan efisien.

CIDR itu bikin ada subnet mask, jadinya IP address bisa dibagi jadi blok, yang ukurannya bisa disesuaikan sendiri. Tambah lagi, kalau ada tabel routing pake CIDR, lebih efisien karena blok yang besar-nya bisa diwakilin sama 1 entry doang.

6. Jelaskan bagaimana cara kerja cookies pada HTTP dalam penanganan sesi web

Saat client request ke server, server bisa balikin response yang berisi cookie, nanti cookie-nya disimpan di browser. Nah nanti setiap kali client request lagi, cookie-nya bakal diikutin ke dalam request-nya, nanti server bisa membaca cookie-nya untuk identifikasi sesi client, jadi ke-track sesinya, sama bisa buat autentikasi juga.

7. Sejumlah besar alamat IP tersedia pada alamat mulai dari 198.16.240.0. Jika ada 4 organisasi besar A, B, C dan D yang meminta alamat sebesar 8000, 1000, 4000, dan

2000 secara berurutan. Jelaskan alokasi IP yang diberikan ke masing-masing organisasi tersebut. Untuk masing-masing organisasi, berikan alamat awal IP yang diberikan, alamat akhir IP, dan network mask dalam bentuk notasi w.x.y.z/s (e.g. 1.2.3.0/22)

Organisasi A (8000 alamat)

2^{13} alamat = 8192 alamat

Network mask = $32 - 13 = 19$ (/19)

Alamat awal = 198.16.240.0

Alamat akhir = $198.16.240.0 + 8192 = 198.17.15.255$ / 19

Organisasi B (1000 alamat)

2^{10} alamat = 1024 alamat

Network mask = $32 - 10 = 22$ (/22)

Alamat awal = 198.17.15.255

Alamat akhir = $198.17.15.255 + 1024 = 198.17.19.255$ / 22

Organisasi C (4000 alamat)

2^{12} alamat = 4096 alamat

Network mask = $32 - 12 = 20$ (/20)

Alamat awal = 198.17.19.255

Alamat akhir = $198.17.19.255 + 4096 = 198.17.35.255$ / 20

Organisasi D (2000 alamat)

2^{11} alamat = 2048 alamat

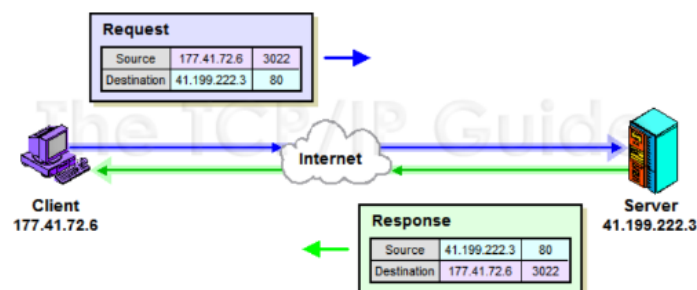
Network mask = $32 - 11 = 21$ (/21)

Alamat awal = 198.17.35.255

Alamat akhir = $198.17.35.255 + 2048 = 198.17.43.255$ / 21

8. Soal

- a. Jelaskan cara kerja dari protokol HTTP sesuai ilustrasi berikut beserta dengan pengalaman yang digunakan



Cara kerja HTTP adalah saat client ingin mengakses halaman web, client mengirim HTTP request ke server, terus nanti sama server dikasih web document-nya, atau resource yang disimpan di URI tersebut. Nah di header HTTP-nya itu baru diisi source sama destination-nya. Kalau response-nya berhasil, nanti ditambahin kode status 200 di header-nya.

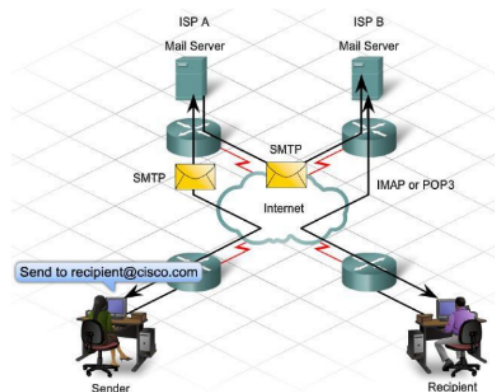
- b. Jelaskan cara kerja dari protokol DNS beserta message-nya

Cara kerja DNS itu saat mengakses web, client bakal ngirim request ke DNS server buat dapetin IP numerik dari nama yang terasosiasi itu

Kalau server DNS gapunya info nya di cache, nanti bakal request ke DNS lain (root DNS, TLD, authoritative DNS) buat nyari, ini namanya DNS lookup

Kalau udah nemu, tinggal dikirim IP numeriknya ke client yang request, baru client bisa send HTTP request ke server dengan IP tersebut

- c. Jelaskan fungsi dari protokol SMTP, IMAP, dan POP3 sesuai ilustrasi berikut ini beserta dengan perbedaan antara ketiganya.



SMTP itu cuma dipake buat ngirim email, bukan buat ngambil atau menyinkronkan, malah kaya antar mail server doang gitu

IMAP itu dipake menyinkronkan email antar server dan client, jadi bisa multidevice access, sama manage message-nya lebih fleksibel

POP3 itu dipake buat nyimpen email-nya ke perangkat terus dihapus dari server, jadi akses-nya terbatas


9. Anda ingin membangun sebuah jaringan untuk keperluan eksperimen di lab berupa akses web terenkripsi (https) tetapi anda tidak ingin mengeluarkan biaya untuk membeli sertifikat digital dari Certification Authority. Terkait masalah tersebut, apa yang anda dapat lakukan agar server web anda tetap bisa diakses secara secure (https) tetapi tidak perlu mengeluarkan biaya?

Bisa dua alternatif, antara menggunakan SSL/TLS certificate gratis, atau membuat self-signed certificate.

Contoh: Menggunakan let's encrypt (CA gratis) atau OpenSSL buat bikin kunci privat dan sertifikat sendiri (self-signed)

Alternatif lainnya, service-nya taruh aja di atas cloudflare, yang udah nyediain HTTPS tanpa perlu kita urusin certificate-nya.

10. Seorang mahasiswa IF3130 melancarkan command nslookup seperti diperlihatkan di bawah ini.

 Select Command Prompt - nslookup

```
C:\Users\User>nslookup
Default Server:  ns2.ITB.ac.id
Address:  167.205.22.123

> www.google.com
Server:  ns2.ITB.ac.id
Address:  167.205.22.123

Non-authoritative answer:
Name:      www.google.com
Addresses:  2404:6800:4003:c04::69
           172.217.194.99
           172.217.194.103
           172.217.194.104
           172.217.194.105
           172.217.194.106
```

- a. Jelaskan apa fungsi command nslookup
Buat dapetin informasi domain name (DNS, domain name system, nslookup itu name system lookup), basically nyari ke DNS server IP yang terasosiasi dengan suatu NS
- b. Apa yang dimaksud dengan Default Server pada gambar di atas?
Default server itu DNS server yang dirujuk sama client kita
- c. Apa yang dimaksud dengan Non-authoritative answer pada gambar di atas
Non-authoritative answer menunjukkan jawaban yang diberikan DNS server itu bukan dari DNS server yang punya otoritas ke domain tersebut. Basically ini info yang benar tapi bukan dari sumbernya langsung