RANGKUMAN UTS JARKOM

Intro

Network terdiri dari node - link - node, kalo node nya cuma 2 berarti gabisa banyak link

Web browser → network app untuk ambil, saji, dan lintas sumber informasi di WWW

Jaringan dibagi 2 berdasarkan **jenis transmisi**:

1. Broadcast links
2. Point-to-point links

Pengelompokan jaringan **berdasarkan skala**:

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

PAN (Personal Area Network) → jarkom untuk komunikasi antar perangkat komputer

LAN (Local Area Network) → jarkom yg mencakup area fisik kecil (rumah, kantor, dll)

MAN (Metropolitan Area Network) → jarkom besar yg mencakup suatu kota/area luas

WAN (Wide Area Network) → jarkom luas bgt mencakup metropolitan, regional, nasional

Elemen network:

1. Hardware
    - Devices
    - Medium → channel untuk message travels
2. Software
    - Message → web page, email, telp, vid, mulmed streaming
    - Rules/Agreement → skema addressing (IP, MAC address, port number), protocol

Network protocol → deretan layer-layer yang memberikan layanan ke layer di atasnya, layer N mesin A berkomunikasi dengan layer N mesin B (peer) dgn protocol

OSI Layered Model (dari yg paling bawah)

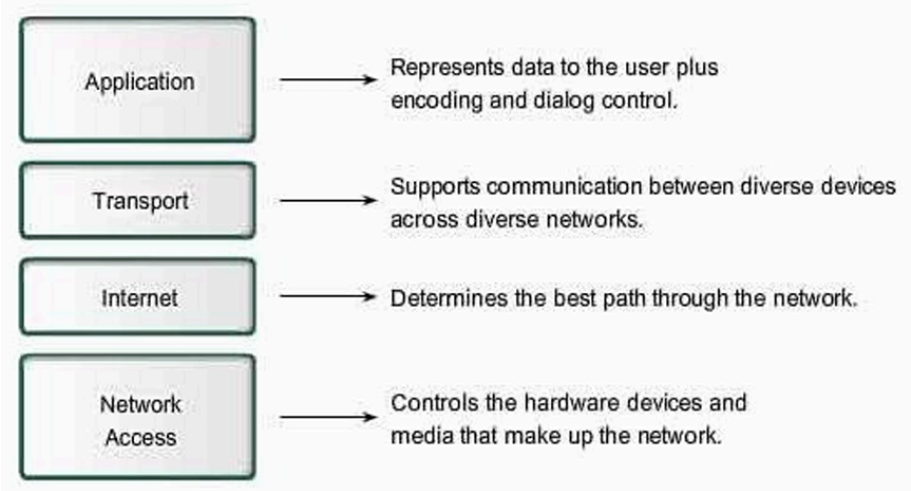
1. Media layers → fokus ke komunikasi (networking)
    a. Physical → bit, mengatur spesifikasi elektrikal, mekanikal, prosedural, dan fungsional untuk aktivasi, pengelolaan, serta deaktivasi sambungan fisik antar end systems, termasuk level tegangan, timing sinyal, data rate fisik, jarak transmisi maksimum, dan tipe konektor.
    b. Data Link **(MAC ADDRESS)** → frame, menyediakan layanan transmisi data bebas error antar dua node yang terhubung melalui physical layer dengan memecah data menjadi frame, mengirimkannya, serta menangani pengakuan frame, deteksi dan koreksi error, kontrol aliran, dan akses medium.
    c. Network **(IP ADDRESS)** → packet, mengatur pengiriman paket dari asal ke tujuan dalam jaringan dengan menentukan rute, congestion control, informasi untuk akuntansi, dan menangani interkoneksi antara subnet yang heterogen.
2. Host layers → fokus ke hosting (basically the nodes)
    a. Transport **(PORT NUMBERS)** → segment, layer end-to-end terendah yang menyediakan end-to-end flow control, end-to-end error detection & correction, serta congestion control tambahan.

b. Session → data, menyediakan dialogue control (giliran mengirim data), token management (akses ke resource bersama), dan sinkronisasi data (status terakhir sebelum link putus).
c. Presentation → data, menangani sintaks dan semantik data yang dikirimkan, menyediakan abstraksi data yang seragam untuk komunikasi antar komputer yang heterogen, yang disent voice, text , image, etc.
d. Application → data, appnya adalah telnet, file transfer, e-mail, newsgroup, web, directory lookup, dan pengambilan informasi.

Terminologi OSI

Elemen aktif di setiap layer disebut **entities** (bisa berupa perangkat keras atau lunak), **entitas pada layer yang sama** di mesin berbeda disebut **peer entities**, data dikirim dalam satuan **Protocol Data Units (PDU)**, dan layer N sebagai **service provider** menyediakan layanan untuk layer N+1 sebagai **service user**.

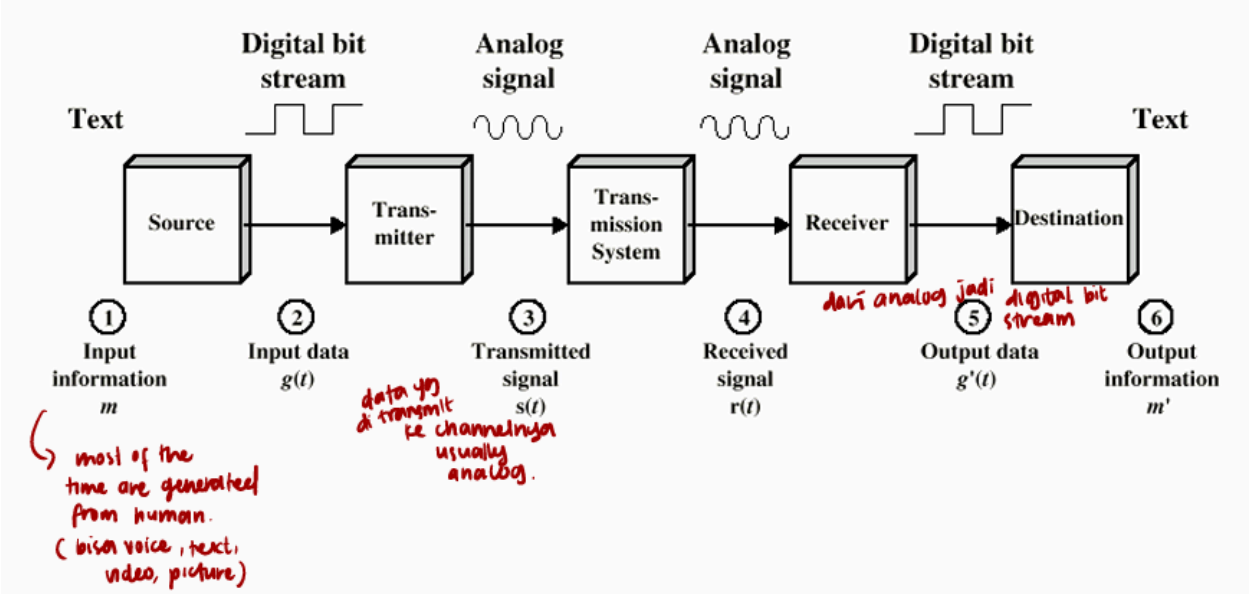
TCP/IP Model



Internet Layered Model

1. Physical → hardware (cables, air)
2. Data Link → software (NIC alias Network Interface Card), protocol ARP, Ethernet
3. Network → software (OS), protocol IP, ICMP, RIP
4. Transport → software (OS), protocol TCP, UDP
5. Application → software (program), protocol HTTP, FTP, DNS, DHCP

Physical Layer



Transmission System

1. **Transmitter** → mengubah **informasi menjadi sinyal yang cocok** untuk transmisi dan mengalirkan energi ke media komunikasi, seperti telepon yang mengubah suara menjadi arus listrik dan modem yang mengubah bit menjadi tones.
2. **Receiver** → menerima energi dari media komunikasi dan mengubah **sinyal yang diterima ke bentuk yang sesuai untuk pengguna**, seperti telepon yang mengubah arus menjadi suara dan modem yang mengubah tones menjadi bit.

Data	**Signal**
Informasi unit bisa disimpan di storage devices	Transmission unit bisa transmitted lewat transmission media

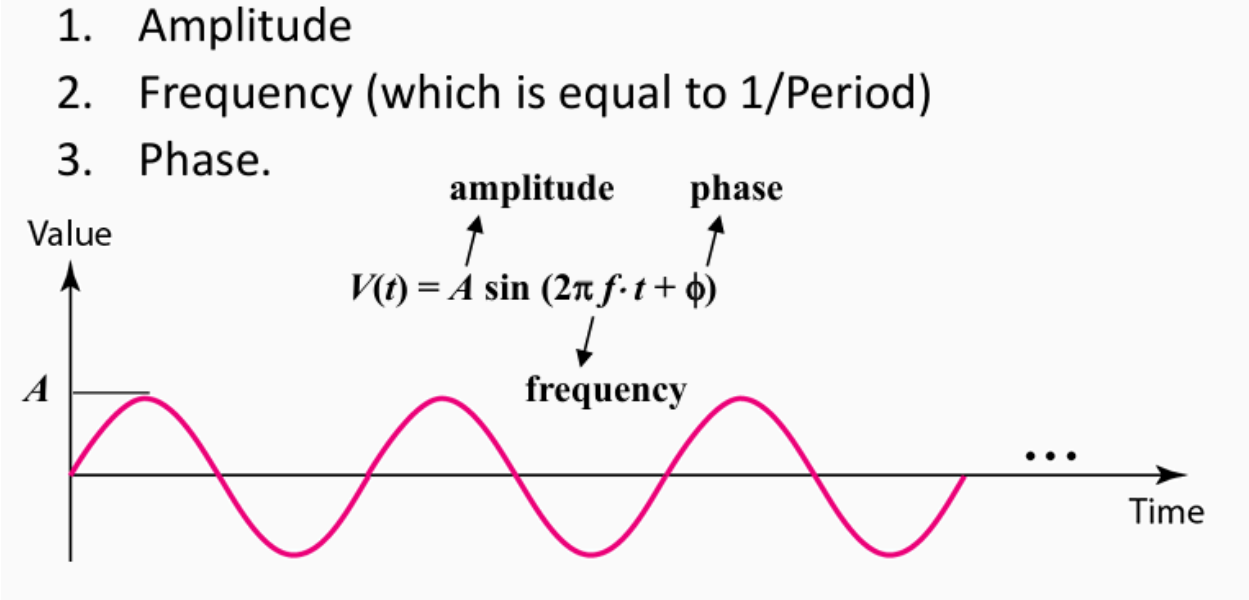
Data menjadi signal di **physical layer**

Analog	**Digital**
Continuous Infinite values	Discrete Limited number of val (tergantung bits)

Analog signal yg wave, digital signal yg kotak

Aperiodic	**Periodic**
Frekuensi sinyal yg akan datang **tidak pasti**	Analog → simple n composite Simple gabisa didecompose, waven ya sine doang Composite bisa didecompose, wave nya bisa macem-macem: square, triangular, sawtooth, fourier series, sine

Karakteristik sine waves



Inget rumus-rumus gelombang la pokonya

T = 1/f f = 1/T

c = λ * f λ = c/f → c = 3 * 10^8

Notes on Frequency

1. Frekuensi adalah laju perubahan sinyal terhadap waktu.
2. Frekuensi **Tinggi** → perubahan **cepat** dalam **waktu singkat**.
3. Frekuensi **Rendah** → perubahan **lambat** dalam rentang **waktu lama**.
4. Frekuensi **Nol** → sinyal yang **tidak berubah**.
5. Frekuensi **Tak Terhingga** → sinyal yang **berubah seketika**.

Bandwith → difference between **highest and lowest freq**

B = fh - fl

bit duration = time period for a bit = 1 / bit rate

bit length = distance a bit occupies the medium = propagation speed × bit duration

Data rate depends on three factors:

- The bandwidth available
- The level of the signals we use
- The quality of the channel (the level of noise)

Nyquist rate

Teori → If we can have infinitely many levels, then we can achieve infinite bit rate

Real → gabisa karena increasing the levels of a signal may reduce the reliability of the system

Bit rate = 2 x bandwidth x log2 L

bandwidth is the analog bandwidth, and, L is the signal level.

Encoding

Langkah pertama dalam membuat node dan link menjadi blok bangunan yang dapat digunakan adalah memahami cara menghubungkannya agar bit dapat ditransmisikan dari satu node ke node lain, dengan mengkodekan data biner sumber ke dalam sinyal yang dapat dibawa link dan mendekodenya kembali di node penerima.

Source Coding (jaringan menangani aliran 0 dan 1)

- Source Encoding → mengompres dan menghilangkan redundansi
- Source Decoding → membalikkan proses tersebut source encoding

Channel Coding (physical transmission link)

- Channel Encoding → menambahkan redundansi untuk transmisi lebih andal sesuai kondisi kanal.
- Channel Decoding → kebalikan dari channel encoding.

- Observasi → source encoding menghilangkan informasi tak berguna, sementara channel encoding menambahkan informasi berguna; keduanya berhubungan dengan redundansi.

Modulation/Demodulation

- Modulation → memetakan bit ke bentuk gelombang atau simbol untuk transmisi yang lebih baik, lalu menggeser ke frekuensi carrier yang diizinkan.
- Demodulation → kebalikan dari modulasi.
- Detection → memulihkan sinyal termodulasi dari sinyal terdistorsi dan berisik yang diterima.

Perbedaan

Jenis	**Tujuan**	**Contoh**
Source Coding	Menghilangkan redundansi untuk penggunaan ruang penyimpanan/ transmisi yang efisien	Huffman coding / Kode Morse
Channel Coding	Pemetaan pra-transmisi untuk memungkinkan koreksi error	Parity check / Hamming code / Reed-Solomon code
Line Coding	Menyesuaikan sinyal dengan karakteristik elektrik dari saluran transmisi	-

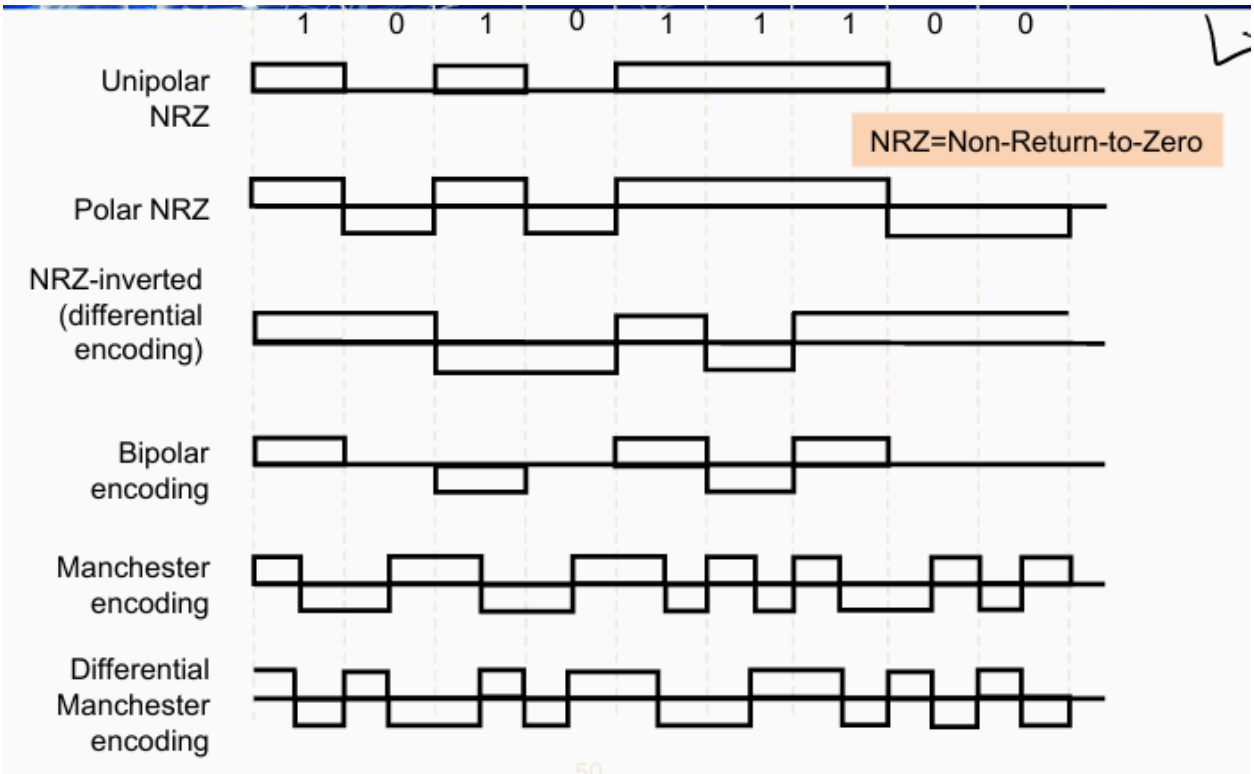
Order: source coding → channel coding → line coding

Line Coding

Mapping biner ke sinyal digital dilakukan agar sesuai dengan kebutuhan sistem, seperti:

- **Daya Transmisi**: Mengontrol konsumsi daya.
- **Sinkronisasi Bit**: Transisi sinyal membantu pemulihan timing.
- **Efisiensi Bandwidth**: Menghindari transisi yang menghabiskan bandwidth.
- **Konten Frekuensi Rendah**: Menghindari sinyal rendah yang bisa menyebabkan “droop.”
- **Deteksi Error**: Kemampuan mendeteksi error sangat membantu.
- **Kompleksitas/Biaya**: Kode harus mudah diimplementasikan di chip berkecepatan tinggi.

Contoh Line Coding



Transmission Medium → jalur fisik/saluran komunikasi yang mengirimkan informasi dari pengirim ke penerima.

Throughput → mengukur kecepatan nyata pengiriman data dalam jaringan; meskipun bandwidth link B bps, throughput T bps mungkin lebih rendah (T ≤ B), misalnya, link 1Mbps mungkin hanya mendukung 200kbps jika perangkat terbatas.

Latency/delay → waktu yang dibutuhkan agar seluruh pesan tiba di tujuan sejak bit pertama dikirim dari sumber.

Latency = propagation time + transmission time + queueing time + processing delay

Intro

Prinsip Dasar

Tantangan utama dalam komunikasi data adalah memastikan **reliabilitas**, mengingat sinyal dapat melemah, terdistorsi, atau terganggu oleh batasan bandwidth. Data yang dikirim bisa rusak atau hilang, sehingga **Data Link Layer** bertugas mengendalikan kesalahan ini dengan pembagian data menjadi frame dan penambahan bit pemeriksa kesalahan.

Peran Data Link Layer

Data Link Layer menangani kontrol error, deteksi error, **flow control**, manajemen link, dan akses medium. Layanan ini menghubungkan layer fisik dengan network layer.

Framing

Data Link Layer membagi aliran bit menjadi frame untuk mempermudah transmisi dan deteksi error. Metode framing termasuk **Byte-Oriented (Sentinel dan Byte Counting → BISYNC, DDCMP, PPP)**, **Bit-Oriented (Bit Stuffing → HDLC)**, dan **Clock-Based (SONET)**.

Error Detection and Correction

Pengendalian error menggunakan bit tambahan untuk mendeteksi dan mengoreksi kesalahan, termasuk metode **Parity Check**, **Checksum**, dan **Cyclic Redundancy Check (CRC)**. Error detection memungkinkan sistem mengetahui kesalahan tanpa mengirim ulang data.

Cyclic Redundancy Check (CRC) pada **Data Link Layer** adalah metode untuk mendeteksi kesalahan dalam transmisi data dengan menambahkan bit tambahan yang dikenal sebagai **Frame Check Sequence (FCS)** pada data asli. Berikut adalah rincian CRC berdasarkan isi dari PPT yang diberikan:

1. **Konsep Dasar CRC**:
    - **CRC** adalah kode deteksi kesalahan yang dirancang khusus untuk menangani **burst errors** (serangkaian bit yang salah). CRC sering digunakan dalam jaringan digital dan perangkat penyimpanan, seperti **hard disk**.

- Untuk menggunakan CRC, pengirim menambahkan **FCS** pada data asli sehingga keseluruhan data dapat dibagi sempurna dengan **pembagi tetap**. Jika data yang diterima tidak memiliki sisa pembagian, maka dianggap bebas kesalahan.

2. **Cara Kerja CRC**:
   - **Pembentukan FCS**:
     - Pengirim mendapatkan data mentah dalam bentuk bit, kemudian melakukan **left shift** atau pergeseran ke kiri sebanyak `(n-k)` bit, di mana `n` adalah total panjang frame (data + FCS) dan `k` adalah panjang data asli.
     - Data yang sudah digeser ini kemudian dibagi dengan **polinomial generator** tertentu yang telah ditentukan (sering disebut G atau `g(x)`), menggunakan aritmatika modulo-2.
     - **Sisa hasil pembagian** ini disebut **FCS** dan ditambahkan ke data asli untuk membentuk frame lengkap yang siap ditransmisikan.
   - **Proses Verifikasi CRC**:
     - Penerima menerima frame yang terdiri dari data dan FCS. Frame ini kemudian dibagi dengan **generator polinomial** yang sama.
     - Jika tidak ada sisa pembagian, penerima menganggap bahwa frame bebas dari kesalahan. Jika ada sisa, maka error terdeteksi, dan frame dianggap rusak.
3. **Matematika di Balik CRC**:
   - **Polinomial**: Representasi data dan FCS dalam bentuk polinomial, misalnya data `1101` bisa dianggap sebagai `x^3 + x^2 + 1`. Generator polinomial juga ditulis dalam bentuk yang serupa, misalnya `x^3 + x + 1`.
   - **Operasi Modulo-2**: CRC menggunakan XOR (exclusive OR) sebagai operasi utama, setara dengan aritmatika **modulo-2**, untuk mengeliminasi atau menghasilkan sisa pembagian.
4. **Kekuatan CRC dalam Error Detection**:
   - CRC sangat efektif dalam mendeteksi **single-bit errors**, **double-bit errors**, **odd number of errors**, dan **burst errors** dengan panjang tertentu.

- Generator polinomial dipilih sedemikian rupa agar dapat mendeteksi pola kesalahan umum. Misalnya, generator `x^16 + x^12 + x^5 + 1` digunakan untuk CRC-16, yang cocok untuk mendeteksi error pada frame dengan panjang tertentu.

5. **Jenis dan Aplikasi CRC**:
   - **CRC-12**: Biasanya digunakan dalam **transmisi karakter 6-bit**, menghasilkan 12-bit FCS.
   - **CRC-16** dan **CRC-CCITT**: Umum untuk transmisi data 8-bit di LAN (misalnya, untuk Ethernet), dan menghasilkan 16-bit FCS.
   - **Implementasi Hardware**: CRC sering diimplementasikan di hardware menggunakan **shift register** untuk efisiensi.
6. **Proses Penghitungan CRC dalam Contoh**:
   - Misalnya, untuk data `1101` dan generator polinomial `x^3 + x + 1`:
     - Data di-extend dengan `n-k` bit menjadi `1101000`.
     - Data ini kemudian dibagi dengan generator menggunakan XOR di setiap langkah hingga menghasilkan sisa atau FCS yang ditambahkan ke data asli.
     - Frame akhir yang berisi data asli dan FCS dikirim, dan penerima akan melakukan pembagian yang sama untuk verifikasi.
7. **Kode CRC Umum**:
   - **CRC-12**: `1 + x + x^2 + x^3 + x^11 + x^12`
   - **CRC-16**: `1 + x^2 + x^15 + x^16`
   - **CRC-CCITT**: `1 + x^5 + x^15 + x^16`

CRC secara keseluruhan adalah metode efisien dan andal untuk menjaga integritas data dalam transmisi, terutama di jaringan dengan risiko tinggi terjadinya gangguan.

Flow Control

Flow control menjaga aliran data stabil antar node dengan metode seperti **Stop-and-Wait ARQ** dan **Sliding Window Protocol**, yang membatasi jumlah frame yang dapat dikirim tanpa ACK untuk mencegah overload.

Ethernet and Medium Access

Ethernet menggunakan **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** untuk mengelola akses ke jaringan dan mencegah tabrakan sinyal. Teknik ini mengutamakan transmisi segera ketika jalur kosong, dan kembali mencoba setelah tabrakan terdeteksi.

MAC Addressing

MAC Address adalah alamat fisik unik yang digunakan untuk mengidentifikasi perangkat di jaringan lokal. Setiap NIC memiliki MAC Address yang berfungsi dalam pengiriman frame antar perangkat di jaringan LAN.

Perkembangan Ethernet

Ethernet berkembang dari **10 Mbps** hingga **10 Gbps**, dengan teknologi seperti **Gigabit Ethernet** untuk jaringan backbone dan **Fast Ethernet** untuk kompatibilitas tinggi dan kecepatan yang lebih cepat.

Challenges and Efficiency of Ethernet

Utilisasi Ethernet optimal pada beban ringan, dengan kapasitas optimal di bawah 30% untuk mengurangi tabrakan. Ethernet ideal untuk biaya rendah, mudah dikelola, tetapi efisiensinya menurun pada jaringan dengan banyak host atau jarak transmisi panjang.

Bridging and Switching

1. Pengantar

- **Internetworking** mengacu pada jaringan yang menghubungkan beberapa LAN untuk memberikan layanan pengiriman paket antar host. Untuk menghubungkan jaringan LAN ke LAN lain atau ke WAN, digunakan perangkat seperti **hub**, **bridge**, **switch**, dan **router**.

2. Topik Utama

- **Interkoneksi Segmen LAN**: Melibatkan perangkat seperti **hub** (Physical Layer), **bridge** (Data Link Layer), dan **Layer 2 Switch**.
- **Interkoneksi Jaringan**: Menggunakan perangkat **Layer 3 Switch** dan **router** di network layer.

3. Hub (Physical Layer)

- **Hub** menghubungkan beberapa node dalam LAN dan beroperasi di Layer 1 (Physical Layer). Hub menyebarkan data ke semua perangkat yang terhubung tanpa filter, sehingga semua perangkat berbagi **domain kolisi yang sama**. Kelemahan dari hub adalah risiko kolisi tinggi dan pemborosan bandwidth.

4. Bridge (Data Link Layer)

- **Bridge** bekerja di Layer 2 (Data Link Layer) dan mengelola lalu lintas melalui filtering berdasarkan **alamat MAC**. Bridge mampu mengisolasi domain kolisi, meningkatkan efisiensi jaringan.
- **Fungsi Bridge** meliputi:
    - **Self-Learning**: Bridge memiliki tabel forwarding yang dibangun secara otomatis dengan mempelajari alamat MAC dari paket yang diterima. Setiap entri dalam tabel memiliki waktu kadaluarsa untuk mencegah masalah jika perangkat berpindah jaringan.
    - **Forwarding dan Filtering**: Bridge hanya meneruskan paket ke segmen yang relevan.

    - **Loop Avoidance**: Masalah loop dalam jaringan dapat diatasi dengan **Spanning Tree Protocol (STP)**.

5. Switch Ethernet (Layer 2)

- Switch pada dasarnya adalah **multi-port bridge** yang memungkinkan lebih banyak port untuk konektivitas. Switch mempelajari alamat MAC dan meneruskan frame hanya ke port yang sesuai.
- **Switching** pada Layer 2 memungkinkan komunikasi simultan antar perangkat, mengurangi kolisi dan meningkatkan throughput.

6. Algoritma Spanning Tree (STA)

- Algoritma Spanning Tree (STA) mencegah **looping** di jaringan dengan menciptakan **pohon tanpa siklus** yang meliputi semua node. **STP** bekerja dengan:
    - Memilih root bridge.
    - Menentukan port mana yang harus diblokir untuk menghilangkan siklus, sementara tetap mempertahankan jalur redundan untuk **reliabilitas** jaringan.
    - Algoritma ini bersifat dinamis, memungkinkan jaringan untuk beradaptasi ketika ada perangkat yang gagal atau berubah.

7. Virtual LAN (VLAN)

- **VLAN** memungkinkan segmentasi logis dari satu LAN fisik menjadi beberapa jaringan virtual. Ini mengurangi lalu lintas broadcast dan memungkinkan isolasi administrasi.
- VLAN diatur melalui **port-based** atau **MAC address-based** dengan tag VLAN sesuai protokol **IEEE 802.1Q**. VLAN juga dapat diperluas ke WAN melalui **Layer 2 tunneling**.

8. Perbedaan antara Hub, Bridge, dan Router

- **Hub**: Perangkat Layer 1 yang hanya mengulang sinyal tanpa pemahaman protokol MAC, sehingga seluruh jaringan yang terhubung ke hub berada dalam satu domain kolisi.

- **Bridge/Layer-2 Switch**: Perangkat Layer 2 yang menyimpan dan meneruskan frame berdasarkan alamat MAC, membagi jaringan menjadi beberapa domain kolisi.
- **Router/Layer-3 Switch**: Perangkat Layer 3 yang meneruskan paket berdasarkan alamat IP, menggunakan data link layer untuk pengiriman hop-to-hop.

9. Switching dan Forwarding di Layer 3 (Network Layer)

- Network layer bertugas **addressing**, **routing**, dan **forwarding** paket antar segmen LAN atau WAN.
- Model layanan pada network layer terbagi dua:
    - **Datagram (Connectionless)**: Paket dikirim secara independen, setiap paket membawa alamat tujuan lengkap.
    - **Virtual Circuit (VC, Connection-Oriented)**: Paket mengikuti jalur yang sama setelah fase set-up, mendukung **Quality of Service (QoS)** dan lebih stabil untuk koneksi jangka panjang.

10. Datagram vs. Virtual Circuit

- **Datagram**: Tidak memiliki fase set-up koneksi, setiap paket membawa alamat lengkap dan diproses secara independen. Ini cocok untuk jaringan yang fleksibel namun memiliki overhead per paket lebih tinggi.
- **Virtual Circuit (VC)**: Ada fase set-up koneksi sebelum pengiriman data, dan semua paket mengikuti jalur tetap. Ini mengurangi overhead per paket dan cocok untuk koneksi yang membutuhkan QoS.

11. Routing dan Switching Berbasis Sumber (Source Routing)

- **Source Routing** adalah metode switching yang memberikan informasi lengkap tentang topologi jaringan di setiap paket. Host sumber menentukan jalur paket melalui jaringan, cocok untuk jaringan yang lebih kecil atau dengan jalur tetap.

Internetworking

1. Introductions

- **Internetworking** adalah konsep menghubungkan beberapa **LAN** yang mungkin tidak kompatibel, seperti Ethernet dan WiFi. Gabungan dari jaringan yang terhubung ini dikenal sebagai **internetwork**, dengan **Internet** sebagai contoh terbaik. Internetworking melibatkan penggunaan perangkat seperti router untuk menghubungkan berbagai jenis jaringan.

2. Structure of the Internet

- Internet adalah kumpulan jaringan yang terhubung secara ad-hoc tanpa topologi terstruktur. Setiap jaringan bisa menggunakan teknologi yang berbeda dan memiliki kapasitas link yang beragam. **Paket data** mengalir dari satu titik ke titik lain melalui beberapa jaringan, dengan menggunakan **router** yang menghubungkan berbagai jaringan. Berbeda dengan jaringan tradisional, Internet memungkinkan paket data mengambil berbagai jalur yang berbeda menuju tujuan.

3. Internet Protocol (IP)

- **IP** (Internet Protocol) adalah protokol utama yang digunakan untuk membangun internetwork yang heterogen dan scalable. IP berjalan di semua node dalam jaringan dan mendefinisikan infrastruktur yang memungkinkan node dan jaringan berfungsi sebagai satu **jaringan logis**. Model layanan IP bersifat **connectionless** dan **best-effort**, yang berarti paket bisa hilang, datang tidak berurutan, atau mengalami duplikasi.

4. IP Service Model

- **Model Layanan IP** adalah sistem pengiriman paket yang tidak bergantung pada koneksi (connectionless). IP menyediakan skema pengalamatan global yang memungkinkan identifikasi semua host dalam jaringan, namun IP tidak menjamin

keandalan, sehingga beberapa paket bisa hilang, terlambat, atau tiba dalam urutan yang salah.

5. IP Addressing

- **Alamat IP** adalah pengenal unik untuk host atau interface router, umumnya berupa format **IPv4** (32-bit) atau **IPv6** (128-bit). Alamat IP ditulis dalam **notasi desimal bertitik** untuk IPv4 (misalnya, 192.168.0.1) dan dalam notasi **hexadecimal** untuk IPv6. Alamat IP dibagi menjadi dua bagian: bagian **jaringan** (network) dan bagian **host**. Semua perangkat dengan bagian jaringan yang sama dapat berkomunikasi tanpa router.

6. Classful IP Addressing

- Metode **classful addressing** membagi alamat IP menjadi beberapa kelas (A, B, C, D, E) dengan ukuran jaringan yang berbeda. Keterbatasannya adalah ketidakefisienan dalam penggunaan ruang alamat, yang mengarah pada kehabisan alamat IP lebih cepat.

7. Classless InterDomain Routing (CIDR)

- **CIDR** memperbaiki kekurangan classful addressing dengan menggunakan **prefix network** yang fleksibel. Alamat dialokasikan dalam blok kontinyu, yang memungkinkan pengelompokan jaringan dan efisiensi dalam routing. Alamat CIDR ditulis dalam format `a.b.c.d/x`, di mana `x` menunjukkan jumlah bit untuk bagian jaringan.

8. Representation of Address Blocks

- CIDR memungkinkan pengelompokan alamat dalam blok dengan format yang dapat dibaca manusia, seperti `a.b.c.d/x`. Ini membantu mengurangi ukuran tabel routing dengan menggabungkan banyak alamat ke dalam satu entri.

9. IP Forwarding

- **IP Forwarding** mengacu pada pengiriman paket IP berdasarkan **alamat tujuan** di dalam tabel routing. Jika alamat berada di jaringan yang sama, paket dikirim langsung. Jika tidak, paket diteruskan ke router berikutnya hingga mencapai tujuan akhir.

10. Host Configurations

- **Konfigurasi IP** di host bisa dilakukan secara manual atau otomatis. Konfigurasi manual rentan terhadap kesalahan, terutama di jaringan besar, sehingga digunakan metode otomatis seperti **Dynamic Host Configuration Protocol (DHCP)**.

11. Dynamic Host Configuration Protocol (DHCP)

- **DHCP** memudahkan perangkat untuk mendapatkan alamat IP secara otomatis dari server DHCP saat bergabung dengan jaringan. DHCP juga mendukung pengguna mobile dan memungkinkan penggunaan kembali alamat IP untuk menghemat ruang.s

12. MAC vs. IP Addresses

- **MAC Address** adalah pengenal unik pada perangkat jaringan yang diberikan oleh pabrik dan digunakan untuk mengirimkan frame dalam LAN. **IP Address** digunakan untuk routing antar jaringan dan bisa diubah sesuai jaringan yang terhubung. Perbedaan utama adalah MAC bersifat **statis** sedangkan IP **dinamis**.

13. ARP (Address Resolution Protocol)

- **ARP** adalah protokol yang memetakan alamat IP ke alamat MAC di jaringan lokal. Proses ARP melibatkan pengiriman **ARP request** untuk mendapatkan alamat MAC dari IP tujuan dan menerima **ARP reply** yang berisi informasi yang dibutuhkan.

14. IP Datagram dan Header Fields

- **IP Datagram** mengandung semua informasi yang diperlukan untuk pengiriman paket, seperti **alamat sumber dan tujuan**, **TTL (Time to Live)**, dan **protocol ID**.

Header IP dibagi dalam beberapa bagian untuk mengidentifikasi dan mengatur paket selama transmisi.

15. Fragmentation dan Reassembly

- **Fragmentasi** terjadi ketika paket melebihi ukuran MTU jaringan. Paket tersebut dipecah menjadi fragmen yang lebih kecil, yang dirakit kembali di tujuan. Ini memungkinkan pengiriman data melalui jaringan yang heterogen tanpa batasan ukuran paket.

16. Internet Control Message Protocol (ICMP)

- **ICMP** adalah protokol bantu untuk IP yang digunakan untuk mengirim pesan kesalahan dan informasi terkait jaringan. Contoh pesan ICMP termasuk "Destination Unreachable" dan "Time Exceeded", yang membantu dalam diagnosis jaringan.

17. Network Address Translation (NAT)

- **NAT** memungkinkan banyak perangkat di jaringan lokal menggunakan satu alamat IP publik, mengatasi kehabisan alamat IPv4, memberikan keamanan tambahan, dan mendukung **TCP load sharing**. Tipe NAT meliputi **Static NAT, Dynamic NAT, PAT (Port Address Translation)**, dan **Port Forwarding**.

18. IPv4 Address Space Crisis dan IPv6

- Alamat IPv4 (32-bit) terbatas, menyebabkan krisis ruang alamat IP. **IPv6** (128-bit) diperkenalkan sebagai solusi, dengan jumlah alamat yang jauh lebih banyak. IPv6 menggunakan format yang lebih sederhana dan memiliki fitur tambahan seperti **Privacy Extensions** dan **Source Routing**.

19. IPv6 Header

- **Header IPv6** lebih sederhana tanpa beberapa field yang ada di IPv4, seperti checksum dan fragmentasi. Pengelolaan fragmentasi di IPv6 dilakukan oleh

endpoint, bukan oleh router, mencerminkan perubahan prioritas di internet modern.

20. Transitioning to IPv6

- Transisi dari IPv4 ke IPv6 memerlukan peningkatan di seluruh jaringan internet. Teknologi transisi seperti **6to4, Teredo**, dan **6rd** memungkinkan paket IPv6 dikirim melalui jaringan IPv4, mendukung kompatibilitas selama masa transisi.

Routing

1. Dasar-Dasar Routing

- Tujuan: Menentukan jalur terbaik (biasanya terpendek) dari sumber ke tujuan dalam jaringan
- Network dimodelkan sebagai graph dengan router sebagai nodes dan link sebagai edges
- Edge cost bisa berupa delay, tingkat kongesti, dll.

2. Paradigma Routing

- Hop-by-hop Routing:
    - Setiap paket berisi alamat tujuan
    - Setiap router memilih next-hop ke tujuan
    - Paket ke tujuan yang sama bisa mengambil jalur berbeda
- Source Routing:
    - Pengirim memilih jalur ke tujuan secara tepat
    - Router meneruskan paket sesuai spesifikasi

3. Distance Vector Routing

- Menggunakan algoritma Bellman-Ford
- Setiap node membuat vector berisi jarak ke semua node lain
- Setiap router berbagi tabel dengan tetangganya
- Masalah:
    - Count-to-infinity problem
    - Slow convergence untuk bad news
- Solusi: Split horizon dan poison reverse

4. Link State Routing

- Setiap router mendistribusikan Link State Packet (LSP) ke semua router
- LSP berisi:
    - ID node pembuat

    - Cost link ke tetangga
    - Sequence number
    - Time-to-live
- Menggunakan algoritma Dijkstra untuk menghitung jalur terpendek
- Lebih cepat converge dibanding Distance Vector

5. Intra-AS vs Inter-AS Routing

- Intra-AS (Interior Gateway Protocols):
    - RIP (Routing Information Protocol)
    - OSPF (Open Shortest Path First)
    - IS-IS
    - EIGRP
- Inter-AS:
    - BGP (Border Gateway Protocol)
    - Menangani routing antar Autonomous Systems

6. OSPF (Open Shortest Path First)

- Menggunakan algoritma Link State
- Fitur lanjutan:
    - Autentikasi untuk keamanan
    - Mendukung multiple same-cost paths
    - Multiple cost metrics untuk berbagai TOS
    - Mendukung unicast dan multicast
    - Hierarchical routing untuk domain besar

7. BGP (Border Gateway Protocol)

- Standard de facto untuk inter-AS routing
- Fitur:
    - Pertukaran informasi reachability antar AS
    - Menggunakan TCP untuk komunikasi
    - Path attributes (AS-PATH, NEXT-HOP)
    - Policy-based routing

- Pesan: OPEN, UPDATE, KEEPALIVE, NOTIFICATION

8. MPLS (Multi-Protocol Label Switching)

- Mempercepat forwarding IP dengan label fixed length
- Label Switched Router (LSR) meneruskan paket berdasarkan nilai label
- Memungkinkan traffic engineering
- Harus co-exist dengan router IP-only

9. Mobile IP

- Memungkinkan konektivitas jaringan saat host berpindah
- Komponen:
    - Home agent di jaringan asal
    - Care-of address saat mobile host berpindah
    - Foreign agent atau temporary IP address
- Menggunakan IP-in-IP tunneling
- Memiliki masalah routing inefficiency yang bisa dioptimasi

UTS 2021/2022

- Gambarkan dan jelaskan proses encapsulation dan decapsulation berdasarkan OSI Reference Model untuk keseluruhan layernya berdasarkan ilustrasi berikut, baik pada sisi pengirim, intermediate device (router), dan penerima. jelaskan juga Protocol Data Unit (PDU) pada setiap layernya.

Ilustrasi yang Anda berikan menunjukkan aliran data dari pengirim melalui router ke penerima. Dalam konteks OSI Reference Model, proses encapsulation dan decapsulation melibatkan penambahan dan penghapusan header pada setiap layer. Berikut ini penjelasan dari setiap layer dan Protocol Data Unit (PDU) yang terlibat:

Sisi Pengirim:

1. **Layer Aplikasi (Layer 7)**
   - **PDU:** Data
   - Data dari aplikasi yang ingin dikirimkan.
2. **Layer Presentasi (Layer 6)**
   - **PDU:** Data

    - Data mungkin dienkripsi, dikompresi, atau diubah formatnya agar sesuai dengan kebutuhan jaringan atau aplikasi penerima.
3. **Layer Sesi (Layer 5)**
    - **PDU:** Data
    - Mengatur koneksi, mengelola pertukaran data antara aplikasi lokal dan jauh.
4. **Layer Transport (Layer 4)**
    - **PDU:** Segmen (TCP) atau Datagram (UDP)
    - Memastikan pengiriman data yang efektif, seperti pengontrolan kesalahan, pengaturan aliran data, dan pengakuran ulang data. TCP menambahkan header yang menyediakan port sumber dan tujuan, serta kontrol aliran dan manajemen kesalahan.
5. **Layer Network (Layer 3)**
    - **PDU:** Paket
    - Menambahkan informasi routing IP termasuk alamat IP pengirim dan penerima. Ini memungkinkan router untuk memutuskan jalur terbaik untuk paket.
6. **Layer Data Link (Layer 2)**
    - **PDU:** Frame
    - Menambahkan header dan trailer yang menyediakan alamat MAC sumber dan tujuan, kontrol kesalahan, dan isyarat pengaturan aliran untuk memastikan pengiriman data yang andal antar node pada jaringan yang sama.
7. **Layer Physical (Layer 1)**
    - **PDU:** Bit
    - Mengubah frame data menjadi bit dan mengirimkannya melalui media fisik seperti kabel atau gelombang radio.

Intermediate Device (Router):

- Router beroperasi pada Layer 3 (Network).
- **Decapsulation:**
    - Mengambil frame, menghilangkan header Layer 2, dan mengekstrak paket.
- **Routing:**
    - Mengevaluasi informasi pada header paket untuk menentukan rute berikutnya.
- **Encapsulation:**
    - Menambahkan header Layer 2 baru berdasarkan informasi interface yang akan digunakan untuk keluar menuju tujuan selanjutnya.

Sisi Penerima:

- Proses decapsulation terjadi dalam urutan terbalik dari encapsulation:
    1. **Layer Physical (Layer 1)**
        - Menerima bit melalui media fisik.
    2. **Layer Data Link (Layer 2)**
        - Mengonversi bit menjadi frame, memeriksa kesalahan, dan mengirim ke layer atas jika frame valid.
    3. **Layer Network (Layer 3)**
        - Menghilangkan informasi routing, memeriksa alamat IP tujuan.

4. **Layer Transport (Layer 4)**
    - Mengkonversi paket kembali menjadi segmen atau datagram dan melakukan proses pengakuran ulang jika diperlukan.
5. **Layer Sesi (Layer 5), Presentasi (Layer 6), dan Aplikasi (Layer 7)**
    - Pengolahan lebih lanjut termasuk dekompresi, dekripsi, dan akhirnya data diteruskan ke aplikasi tujuan.

Proses ini menjamin bahwa data yang dikirim dari satu aplikasi di satu node berhasil diterima oleh aplikasi yang sesuai di node lain, melintasi jaringan kompleks yang terdiri dari berbagai media transmisi dan teknologi jaringan.

- Error Detection
    a. Pada pendeteksian error menggunakan parity bit, diterima deretan bit 00101110. Tentukan apakah data tersebut mengandung error jika menggunakan: 1) odd parity dan 2) even parity. Jelaskan alasannya.

Pendeteksian Error dengan Parity Bit

Pendeteksian error menggunakan parity bit dilakukan dengan menambahkan sebuah bit (parity bit) pada akhir deretan bit yang dikirim, dengan tujuan untuk membuat jumlah bit yang bernilai '1' menjadi genap (even parity) atau ganjil (odd parity). Mari kita evaluasi data `00101110`: **Menghitung jumlah bit yang bernilai '1'**: Dalam deretan bit `00101110`, ada empat bit '1'.

1) Odd Parity

- **Cara Kerja**: Odd parity memastikan bahwa jumlah total bit '1' termasuk parity bit adalah ganjil.
- **Evaluasi**: Tanpa mengetahui nilai parity bit, kita melihat bahwa jumlah bit '1' yang ada sudah genap (4). Untuk memenuhi odd parity, parity bit harus bernilai '1' untuk membuat jumlah bit '1' menjadi ganjil (5).
- **Kesimpulan**: Jika parity bit adalah '1', maka tidak ada error; jika parity bit adalah '0', maka terdapat error.

2) Even Parity

- **Cara Kerja**: Even parity memastikan bahwa jumlah total bit '1' termasuk parity bit adalah genap.
- **Evaluasi**: Dengan jumlah bit '1' yang sudah genap (4), parity bit harus bernilai '0' agar jumlah bit '1' tetap genap (4).
- **Kesimpulan**: Jika parity bit adalah '0', maka tidak ada error; jika parity bit adalah '1', maka terdapat error.

a. Pada pendeteksian error dengan menggunakan metode CRC, tentukan apakah data yang diterima berikut ini mengandung error atau bebas error. Data yang diterima adalah 11110111110010, lalu pendeteksian CRC menggunakan generator polynomial 10011.

   CRC adalah metode yang lebih kuat untuk mendeteksi kesalahan dibandingkan dengan parity bit. CRC menggunakan pembagian polinomial biner dari data yang dikirim ditambah dengan bit-bit redundan yang diturunkan dari pembagian tersebut. Untuk mengecek error, data yang diterima (termasuk bit-bit redundan) dibagi dengan polynomial generator yang sama, dan jika sisa hasil bagi (remainder) adalah nol, maka tidak ada error yang terdeteksi.

   - **Data yang Diterima**: `11110111110010`
   - **Generator Polynomial**: `10011`

   Langkah-langkah untuk pengecekan menggunakan CRC adalah sebagai berikut:

   1. Tambahkan (n-1) bit '0' di mana n adalah derajat polynomial generator ke data yang akan dikirim. Namun, karena data yang kita terima termasuk bit CRC, kita hanya perlu membaginya dengan polynomial generator.
   2. Lakukan pembagian binary data yang diterima dengan generator polynomial.
   3. Periksa sisa hasil bagi:
   - Jika remainder adalah '0', maka data bebas error.
   - Jika remainder bukan '0', maka data mengandung error.

3. Diketahui deretan bit input: 01001101 11001110
   a. Gambarkan encoding dari bit input jika menggunakan NRZ dan NRZI (start HIGH)
   b. Gambarkan encoding dari bit input jika menggunakan Manchester dengan pola clock berikut:

4. Suatu perusahaan IT di Indonesia diberikan alokasi IP address oleh APNIC. Salah satu IP address dalam alokasi tersebut adalah 180.1.100.150 dengan subnet mask 255.255.255.128. Sebagai seorang network engineer, buatlah rancangan alokasi IP address untuk perusahaan tersebut sesuai dengan prinsip subnetting dan VLSM, dengan kebutuhan tiap divisi antara lain:
   - Marketing (15 host)
   - Operation and Maintenance (50 host)
   - Finance (50 host)
   - Human Resource (4 host)

   Sebagai guideline, lakukan identifikasi parameter IP address berikut pada setiap divisinya:
   a. Prefix-length dan subnet mask
   b. Subnet/network address
   c. Broadcast address
   d. IP address untuk host pertama dan terakhir

Untuk merancang alokasi IP address menggunakan VLSM (Variable Length Subnet Masking), kita akan mulai dengan mengalokasikan subnet dengan kebutuhan host terbanyak terlebih dahulu. Hal ini membantu memastikan bahwa penggunaan alamat IP lebih efisien. IP address yang kita mulai adalah 180.1.100.150 dengan subnet mask 255.255.255.128, yang menunjukkan bahwa ini adalah bagian dari subnet yang lebih besar dengan network address 180.1.100.128/25 (prefix-length 25).

Langkah 1: Identifikasi Besar Subnet untuk Setiap Divisi

1. **Operation and Maintenance: 50 host**
2. **Finance: 50 host**
3. **Marketing: 15 host**
4. **Human Resource: 4 host**

Langkah 2: Tentukan Subnet Mask dan Range IP untuk Setiap Divisi

Operation and Maintenance

- **Hosts**: 50
- **Subnet mask yang diperlukan**: 255.255.255.192 (yang mendukung hingga 62 host, 64 - 2 untuk network dan broadcast address)
- **Prefix-length**: /26
- **Network Address**: 180.1.100.128/26
- **Broadcast Address**: 180.1.100.191
- **IP Host Pertama**: 180.1.100.129
- **IP Host Terakhir**: 180.1.100.190

Finance

- **Hosts**: 50
- **Subnet mask yang diperlukan**: 255.255.255.192
- **Prefix-length**: /26
- **Network Address**: 180.1.100.192/26
- **Broadcast Address**: 180.1.100.255
- **IP Host Pertama**: 180.1.100.193
- **IP Host Terakhir**: 180.1.100.254

Marketing

- **Hosts**: 15
- **Subnet mask yang diperlukan**: 255.255.255.240 (yang mendukung hingga 14 host, 16 - 2)
- **Prefix-length**: /28
- **Network Address**: 180.1.100.0/28
- **Broadcast Address**: 180.1.100.15
- **IP Host Pertama**: 180.1.100.1
- **IP Host Terakhir**: 180.1.100.14

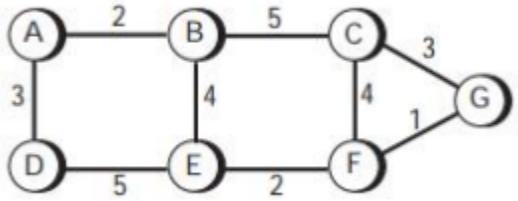
Human Resource

- **Hosts**: 4
- **Subnet mask yang diperlukan**: 255.255.255.248 (yang mendukung hingga 6 host, 8 - 2)
- **Prefix-length**: /29
- **Network Address**: 180.1.100.16/29
- **Broadcast Address**: 180.1.100.23
- **IP Host Pertama**: 180.1.100.17
- **IP Host Terakhir**: 180.1.100.22

Langkah 3: Verifikasi dan Penyesuaian Jaringan

- Semua alokasi subnet harus berada dalam range 180.1.100.0 hingga 180.1.100.255.
- Periksa apakah ada overlap antar subnet atau tidak, dan pastikan bahwa semua host dalam divisi mendapat cukup ruang alamat.

5. Routing



a. Jelaskan proses pembentukan tabel routing dari router/node A jika menggunakan algoritma Bellman-Ford (Distance Vector)
b. Gambarkan dan jelaskan proses pembentukan tabel routing dari router/node C jika menggunakan algoritma Djikstra (Link State)

1. Algoritma Bellman-Ford (Dari Node A)

Algoritma Bellman-Ford adalah metode Distance Vector yang digunakan untuk menghitung jalur terpendek dari satu node ke semua node lain dalam suatu jaringan. Node mengirimkan vector jaraknya kepada tetangganya dan memperbarui vector jaraknya berdasarkan informasi yang diterima.

Langkah-langkah Algoritma Bellman-Ford dari Node A:

- **Inisialisasi**: Node A menginisialisasi jarak ke diri sendiri sebagai 0 dan ke node lainnya sebagai tak hingga (∞), kecuali ke node tetangganya.
- **Iterasi 1**:
    - A ke B: 2
    - A ke D: 3
    - A ke node lain: ∞
- **Iterasi 2**:
    - A belajar dari B bahwa B ke C adalah 5, maka A ke C dapat menjadi 2 (A ke B) + 5 (B ke C) = 7.

    - A belajar dari D bahwa D ke E adalah 5, maka A ke E dapat menjadi 3 (A ke D) + 5 (D ke E) = 8.
- **Iterasi 3**:
    - A belajar dari E (melalui D) bahwa E ke F adalah 2, maka A ke F bisa menjadi 8 (A ke E) + 2 (E ke F) = 10.
    - A belajar dari C bahwa C ke F adalah 4, maka A ke F bisa menjadi lebih pendek, yaitu 7 (A ke C) + 4 (C ke F) = 11 (Namun 10 lebih pendek).
    - A belajar dari C bahwa C ke G adalah 3, maka A ke G bisa menjadi 7 (A ke C) + 3 (C ke G) = 10.
- **Iterasi Berikutnya**:
    - Tidak ada perubahan lebih lanjut dalam jarak.

Tabel Routing Akhir di A:

Destination	Cost	Next Hop
B	2	B
C	7	B
D	3	D
E	8	D
F	10	D
G	10	B

2. Algoritma Dijkstra (Dari Node C)

Algoritma Dijkstra adalah metode Link State yang digunakan untuk menemukan jalur terpendek dari satu node ke semua node lain dengan cara menyimpan informasi tentang seluruh jaringan.

Langkah-langkah Algoritma Dijkstra dari Node C:

- **Inisialisasi**: C menginisialisasi jarak ke diri sendiri sebagai 0 dan ke node lainnya sebagai tak hingga (∞), kecuali ke node tetangganya.
- **Iterasi 1**:
    - C ke B: 5
    - C ke F: 4
    - C ke G: 3
- **Iterasi 2**:
    - C ke A melalui B: 5 (C ke B) + 2 (B ke A) = 7.
    - C ke E melalui B: 5 (C ke B) + 4 (B ke E) = 9.
- **Iterasi 3**:
    - C ke D melalui B: 7 (C ke A) + 3 (A ke D) = 10.
- **Iterasi Berikutnya**:

- Tidak ada perubahan lebih lanjut dalam jarak.

Tabel Routing Akhir di C:

Destination	Cost	Next Hop
A	7	B
B	5	B
D	10	B
E	9	B
F	4	F
G	3	G

6. Konfigurasi host dapat dilakukan secara otomatis dengan menggunakan Dynamic Host Configuration Protocol (DHCP). Host yang baru tersambung ke jaringan akan menghubungi DHCP server di jaringan yang sama. Kedua entitas tersebut akan menggunakan empat messages untuk konfigurasi host.
   a. Jelaskan mekanisme umum dari DHCP dan tugas dari DHCP server.
   b. Jelaskan keempat messages yang digunakan oleh protokol DHCP (nama, unicast/broadcast, penjelasan singkat)
   c. Buatlah signaling ladder diagram untuk konfigurasi host.

Mekanisme Umum DHCP

Dynamic Host Configuration Protocol (DHCP) adalah protokol jaringan yang digunakan untuk secara otomatis memberikan konfigurasi IP kepada host dalam jaringan. DHCP memungkinkan host untuk memperoleh parameter konfigurasi IP yang penting tanpa administrasi manual, menjadikannya sangat berguna dalam jaringan besar dengan banyak perangkat yang sering berganti-ganti atau bergabung ke jaringan.

Tugas dari DHCP Server

Tugas utama DHCP server adalah:

- **Alokasi Alamat IP**: DHCP server mengelola pool alamat IP dan mengalokasikan alamat kepada host yang memintanya.
- **Konfigurasi Parameter Jaringan**: Selain alamat IP, DHCP server juga menyediakan informasi lain seperti subnet mask, default gateway, DNS server addresses, dan lain-lain.
- **Pengelolaan Sewa Alamat IP**: DHCP server mengelola durasi sewa alamat IP yang diberikan kepada host. Ketika sewa mendekati berakhir, host dapat memperbarui sewa atau melepaskan alamat IP jika tidak lagi diperlukan.
- **Menanggapi Permintaan**: DHCP server menanggapi permintaan dari host dan dapat meng-update konfigurasinya berdasarkan kebutuhan.

Keempat Messages yang Digunakan oleh Protokol DHCP

1. **DHCPDISCOVER (Broadcast)**
    - **Deskripsi**: DHCP client mengirimkan DHCPDISCOVER message secara broadcast (ke alamat 255.255.255.255) untuk mencari DHCP server yang tersedia.
    - **Tujuan**: Mencari DHCP server yang bisa melayani permintaan IP.
2. **DHCPOFFER (Broadcast/Unicast)**
    - **Deskripsi**: DHCP server yang menerima DHCPDISCOVER akan merespons dengan DHCPOFFER message, yang menyediakan alamat IP yang tersedia dari poolnya kepada client.
    - **Tujuan**: Menawarkan konfigurasi IP dan parameter jaringan lainnya kepada client.
3. **DHCPREQUEST (Broadcast)**
    - **Deskripsi**: Setelah menerima DHCPOFFER, DHCP client akan mengirimkan DHCPREQUEST message secara broadcast untuk mengindikasikan penerimaan tawaran dari satu DHCP server dan menolak tawaran dari DHCP server lainnya.
    - **Tujuan**: Mengkonfirmasi penerimaan tawaran dan memberi tahu server lain bahwa tawaran tersebut telah diterima oleh client.
4. **DHCPACK (Unicast)**

- **Deskripsi**: Setelah menerima DHCPREQUEST, DHCP server mengirim DHCPACK message secara unicast ke client. Message ini menandakan bahwa alamat IP dan konfigurasi jaringan telah resmi dialokasikan ke client.
- **Tujuan**: Konfirmasi akhir yang menyatakan bahwa alamat IP dan parameter jaringan lainnya telah dialokasikan dan disetujui untuk digunakan oleh client.

Signaling Ladder Diagram untuk Konfigurasi Host dengan DHCP

Berikut adalah ladder diagram yang menggambarkan interaksi antara DHCP client dan DHCP server:

lua

Copy code

```
Client                          Server

  |                               |

  |---- DHCPDISCOVER (Broadcast) ---------------->|

  |<------------------- DHCPOFFER (Broadcast) -----|

  |---- DHCPREQUEST (Broadcast) ------------------>|

  |<------------------- DHCPACK (Unicast) ---------|

  |                               |
```

- **Langkah 1**: Client mengirim DHCPDISCOVER untuk mencari server.
- **Langkah 2**: Server merespons dengan DHCPOFFER yang menawarkan konfigurasi.
- **Langkah 3**: Client memilih konfigurasi dan mengirimkan DHCPREQUEST untuk mengonfirmasi penerimaan.
- **Langkah 4**: Server mengirimkan DHCPACK untuk menyelesaikan konfigurasi.

Proses ini memastikan bahwa konfigurasi IP dilakukan dengan cepat dan efisien, memungkinkan host untuk segera berkomunikasi di jaringan tanpa perlu intervensi manual.