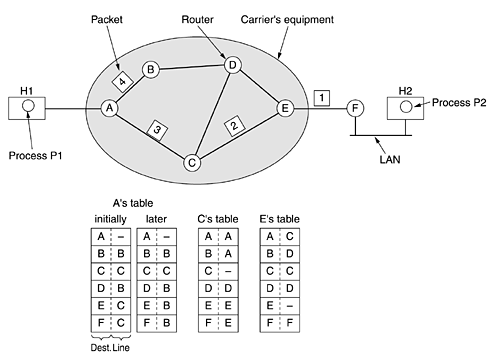
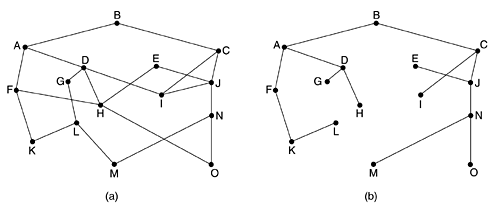
**Network Layer**

* Fungsi utama network layer : mengatur cara pengiriman paket dari sumber sampai ke tujuan.
* Packet Swithing
  + Konsep utama :
    - Store and forward : host sumber akan mengirimkan paket ke router yang terdekat. Di router ini paket akan disimpan sampai lengkap, diperiksa checksumnya, dan dikirim ke router selanjutnya(tergantung algoritma routing). Proses ini akan berlanjut terus sampai paket tiba di host tujuan.
* Servis yang harus disediakan network layer (untuk transport layer) :
  + Servis harus independen terhadap teknologi router.
  + Transport layer tidak perlu mengetahui tipe, jumlah, dan topologi dari router yang digunakan.
  + Network address harus uniform dalam subnet pengantar.
* Tipe servis :
  + Connectionless Service
    - Error control, flow control, dan packet ordering tidak perlu dilakukan pada tiap router.
    - Tidak perlu menginisialisasi koneksi sebelum melakukan komunikasi.
    - Routing table selalu berubah sesuai dengan keadaan saat itu.
    - Paket disebut datagram. Subnet disebut datagram subnet.
    - Kelebihan :
      * Saat ada router yang down, selama ada jalur lain menuju tujuan, koneksi tetap bisa dilakukan.
      * Penyadapan data lebih sulit dilakukan.
    - Kelemahan :
      * Lebih lama dari Connection-Oriented Service.
    - Contoh : Internet



Gambar 1. Routing pada datagram subnet

* + Connection-Oriented Service
    - Harus menginisialisasi koneksi sebelum melakukan komunikasi (disebut menggunakan virtual circuit).
    - Subnet disebut virtual circuit subnet.
    - Routing table ditentukan pada waktu awal koneksi.
    - Paket lebih ringkas karena addressing cukup hanya menyatakan informasi established connection.
    - Kelebihan :
      * Kualitas koneksi lebih baik dari Connectionless Service.
    - Kelemahan :
      * Saat ada router jalur yang down, koneksi putus.
      * Penyadapan data lebih mudah dilakukan.
    - Contoh : ATM
* Algoritma routing :
  + Bagian dari network layer yang bertanggung jawab untuk menentukan jalur keluaran yang harus ditempuh paket yang masuk dalma suatu router.
  + Parameter penting :
    - Correctness : Algoritma harus benar.
    - Simplicity : Algoritma harus sesederhana mungkin.
    - Robustness : Algoritma ini harus tahan uji dan tidak mengalami masalah dalam jangka waktu implementasi yang lama.
    - Stability : Adanya masa stabil setelah masa setup selesai.
    - Fairness : Setiap router harus diberikan jatah yang adil.
    - Optimality : Algoritma dapat menentukan router mana yang harus diprioritaskan.
    - Fairness dan Optimality selalu saling tarik menarik. Algoritma yang baik adalah algoritma yang dapat memberikan kemampuan yang seimbang diantara keduanya.
  + Prinsip optimalitas :
    - Jika router J ada pada jalur optimal dari I menuju K, maka jalur J menuju K yang optimal terletak di jalur itu pula.
    - Sink tree : himpunan rute optimal dari suatu sumber ke semua titik destinasi.



Gambar 2. (a) Graf mula-mula (b) Sink Tree

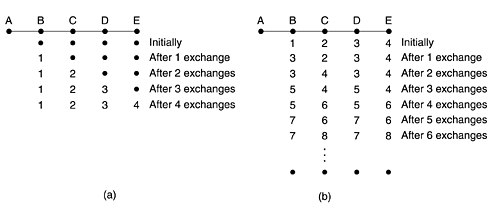
* + Jenis-jenis Algoritma Routing:
    - Non-Adaptive / Static Routing
      * Shortest Path Routing
        + Mencari jalur yang optimal dengan membandingkan parameter dari jaringan (jarak kabel, lama delay, jumlah hop).
        + Perhitungan routing table hanya dilakukan pada setup koneksi.
        + Yang umum digunakan adalah algoritma Djikstra.
      * Flooding
        + Setiap paket yang masuk akan dikirim ke semua jalur keluaran.
        + Paket mungkin kembali lagi ke router yang pernah mengirimnya melalui jalur yang berbeda, oleh karena itu, setiap router harus bisa mengingat paket mana saja yang pernah dikirim olehnya.
        + Dalam teknik selective flooding, forwarding paket dibatasi dalam berapa kali hop yang besarnya ditentukan oleh user (biasanya sebesar diameter graf).
        + Sering digunakan di militer untuk memastikan bahwa jika ada jalur yang rusak, pesan tetap sampai ke tujuan.
        + Menghasilkan delay paling rendah.
    - Adaptive
      * Distance Vector Routing
        + Setiap router menyimpan routing table berisi jarak ke tiap titik (bisa berupa hop, jarak fisik, delay) ke setiap destinasi dan jalur mana yang dibutuhkan untuk mencapai tujuan.
        + Algoritma yang umum digunakann adalah Bellman Ford
        + Digunakan pada ARPA-Net
        + Memiliki masalah Count to Infinity

Masalah ini terjadi jika ada router yang mati.

Routing table akan menghitung jarak menuju router tersebut hingga mencapai kapasitas maksimal.

Disebabkan karena perhitungan algoritma butuh “waktu menjalar”.

Visualisasi :



Gambar 3. (a) Pembuatan routing table titik A (b) Masalah count to infinity

* + - * Link state routing
        + Pengembangan dari Distance Vector Routing.
        + Tidak memiliki masalah count to infinity.
        + Ide dasar :

Setiap router harus dapat mengetahui network address tetangganya.

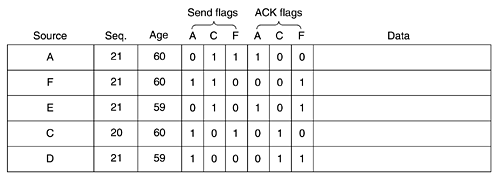
Setiap router harus dapat mengukur biaya pengiriman paket ke tetangganya.

Setiap router harus dapat mengkonstruksi paket mengenai pengetahuan router tersebut.

Setiap router harus dapat mengirim paket ini ke semua router yang lain.

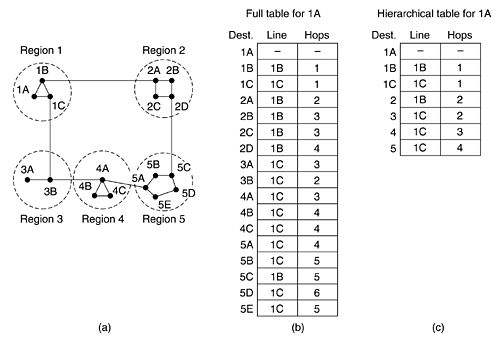
Setiap router harus dapat menghitung jalur tercepat menuju ke router lain.

* + - * + Pencarian biaya menggunakan fungsi ECHO yang akan menghasilkan lama perjalanan paket.
        + Informasi dari tiap router, disebarkan menggunakan link state packet.
        + Setiap router memiliki link state buffer yang akan diupdate setiap mendapatkan link state packet.
        + Link state packet biasanya dikirim menggunakan cara flooding.



Gambar 4. Struktur link packet buffer

* + - * Hierarchical routing
        + Network sekarang menjadi sangat kompleks.
        + Tidak mungkin menyimpan informasi mengenai semua router lain dalam routing table.
        + Network dibagi menjadi grup-grup kecil.
        + Masing-masing grup memiliki metoda pengiriman sendiri-sendiri



Gambar 5. (a) Network (b) Routing Table biasa (c) Hierarchical Routing Table

* + Aplikasi routing :
  + Broadcasting Routing
    - Broadcasting : Mengirimkan pesan ke banyak hosts.
    - Ada beberpaa kemungkinan routing algorithm yang dipakai :
      * Mengirimkan individual packet ke masing-masing hosts
        + Paket yang dikirim banyak.
        + Memakan banyak bandwith.
        + Tidak efektif.
      * Menggunakan flooding
        + Paket yang dikirim banyak.
        + Memakan banyak bandwith.
        + Tidak efektif.
      * Multidestination routing
        + Setiap paket mengandung list dari host yang akan dikirimkan.
        + Setiap router akna memeriksa list tersebut dan mengirimkan sesaui dengan kebutuhan.
      * Spanning tree routing
        + Harus mencari spanning tree dulu, dan harus disimpan di tiap router. Boros memori.
        + Diperbaiki dengan reverse path forwarding

Memeriksa apakah paket yang diterima berasal dari titik yang biasanya digunakan oleh sumber broadcast untuk mengirim.

Jika ya, kirim paket ke semua jalur kecuali jalur masuk (seperti flooding). Jika tidak, discard paket.