JARINGAN KOMPUTER

Untuk Pemula

Fauzan Prasetyo Eka Putra, M.Kom.
Muhammad Umar Mansyur

litrus.

Fauzan Prasetyo Eka Putra, M.Kom.
Muhammad Umar Mansyur

JARINGAN KOMPUTER

Untuk Pemula

Penerbit Litnus.

JARINGAN KOMPUTER UNTUK PEMULA

Ditulis oleh:
Fauzan Prasetyo Eka Putra, M.Kom.
Muhammad Umar Mansyur

Diterbitkan, dicetak, dan didistribusikan oleh
PT. Literasi Nusantara Abadi Grup
Perumahan Puncak Joyo Agung Residence Kav. B11 Merjosari
Kecamatan Lowokwaru Kota Malang 65144
Telp : +6285887254603, +6285841411519
Email: literasinusantaraofficial@gmail.com
Web: www.penerbitlitnus.co.id

Cetakan I, Maret 2023

Perancang sampul: Syafri Imanda
Penata letak: Syafri Imanda

ISBN : 978-623-8227-44-0
xvi + 170 hlm. ; 15,5x23 cm.

KATA PENGANTAR

“Dengan penuh kebahagiaan, kami mempersembahkan buku ini untuk para profesional dan pelajar bidang teknologi informasi. Dalam buku ini, kami akan membahas tentang dunia jaringan komputer, sebuah bidang yang sangat penting dan berkembang pesat dalam era digital saat ini.

Jaringan komputer merupakan dasar dari segala aktivitas online dan memegang peran yang sangat penting dalam mengatur dan menyediakan akses informasi bagi masyarakat. Oleh karena itu, penting bagi kita untuk memahami bagaimana jaringan komputer bekerja dan bagaimana mengatasi masalah yang muncul dalam implementasi jaringan.

Buku ini dirancang untuk memberikan pemahaman dasar tentang jaringan komputer dan membahas topik-topik yang penting dalam bidang ini, seperti protokol jaringan, keamanan jaringan, dan implementasi jaringan. Kami berharap bahwa buku ini dapat menjadi sumber informasi yang berguna bagi para pemula dan profesional yang ingin memperdalam pengetahuan mereka tentang jaringan komputer.

Kami berharap buku ini dapat membantu Anda memahami jaringan komputer dengan lebih baik dan membantu Anda dalam mempersiapkan diri untuk menghadapi tantangan dan peluang di bidang teknologi informasi. Selamat membaca”.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
JARINGAN KOMPUTER DAN INTERNET

Internet saat ini bisa dibilang adalah sistem rekayasa terbesar yang pernah dibuat oleh umat manusia, dengan ratusan jutaan komputer yang terhubung, tautan komunikasi, dan sakelar; dengan miliaran pengguna yang terhubung melalui laptop, tablet, dan smartphone; dan dengan berbagai hal baru yang terhubung dengan Internet termasuk konsol game, sistem pengawasan, jam tangan, kacamata, termostat, timbangan badan, dan mobil. Jadi pada pembahasan kali ini kita akan belajar bahwa Internet adalah jaringan dari jaringan, dan kita akan pelajari bagaimana jaringan ini terhubung satu sama lain.

A. Apa itu Internet?

Dalam buku ini, kita akan menggunakan Internet publik, jaringan komputer tertentu, untuk prinsip utama kita mendiskusikan jaringan komputer dan protokolnya. Tapi apa itu Internet? Ada beberapa cara untuk menjawab pertanyaan ini. Pertama, kita dapat menggambarkan mur dan baut Internet, yaitu dasar komponen perangkat keras dan perangkat lunak yang membentuk Internet. Kedua, kita bisa menggambarkan Internet di syarat infrastruktur jaringan yang menyediakan layanan untuk aplikasi terdistribusi.

1. Deskripsi Mur dan Baut

Sistem akhir dihubungkan oleh jaringan tautan komunikasi dan sakelar paket. Jenis tautan komunikasi, yang terdiri dari berbagai

jenis media fisik, termasuk kabel koaksial, kawat tembaga, serat optik, dan spektrum radio. Tautan yang berbeda dapat mengirimkan data pada kecepatan yang berbeda, dengan laju pengiriman tautan yang diukur bit / detik. Ketika satu sistem ujung memiliki data untuk dikirim ke sistem ujung lain, sistem akhir mengirim segmen data dan menambahkan byte header ke setiap segmen. Paket informasi yang dihasilkan, dikenal sebagai paket dalam jargon jaringan komputer, kemudian dikirim melalui jaringan Internet dengan tujuan sistem akhir, di mana mereka dipasang kembali ke dalam data asli.

Sistem akhir, sakelar paket, dan bagian lain dari Internet protokol yang mengontrol pengiriman dan menerima informasi dalam Internet. Transmission Control Protocol (TCP) dan Internet Protocol (IP) adalah dua protokol paling penting di Internet. Protokol IP menentukan format paket yang dikirim dan diterima di antara router dan sistem akhir. Internet protokol utama secara kolektif dikenal sebagai TCP / IP.

2. Deskripsi Services (Layanan)

Internet juga sebagai infrastruktur yang menyediakan *layanan ke aplikasi*. Selain aplikasi tradisional seperti e-mail dan penjelajahan Web, aplikasi internet termasuk aplikasi smartphone dan tablet, termasuk perpesanan Internet, pemetaan dengan informasi lalu lintas dalam waktu nyata, streaming musik dari cloud, streaming film dan televisi, jejaring sosial online, konferensi video, game multi-player, dan rekomendasi berbasis lokasi sistem. Aplikasi tersebut dikatakan **aplikasi terdistribusi**, karena melibatkan banyak ujung sistem yang saling bertukar data. Yang penting, aplikasi Internet pada sistem akhir tidak berjalan di switch paket dalam inti jaringan. Meskipun packet switch memfasilitasi pertukaran data antara sistem akhir, tidak peduli dengan aplikasi yang merupakan sumber atau tenggelamnya data.

Sistem akhir yang terhubung ke Internet menyediakan antarmuka soket yang menentukan bagaimana suatu program berjalan pada satu sistem akhir yang meminta infrastruktur Internet untuk mengirimkan data ke tujuan tertentu pada program yang berjalan pada sistem ujung yang lain. Antarmuka soket Internet ini adalah seperangkat aturan yang dikirim oleh program harus mengikuti sehingga Internet dapat mengirimkan data ke program tujuan. Internet memiliki antarmuka

soket yang harus diikuti program untuk mengirim data agar Internet mengirimkan data ke program yang akan menerima data.

3. Apa itu Protokol?

› Analogi Manusia

Untuk mudah memahami gagasan protokol jaringan komputer, terlebih dahulu kita mempertimbangkan beberapa analogi manusia, karena kita manusia mengeksekusi protokol sepanjang waktu. Protokol manusia menyatakan bahwa seseorang pertama kali menawarkan salam "Hai" untuk memulai komunikasi dengan orang lain. Secara implisit, seseorang kemudian mengambil respon "Hai" dengan ramah sebagai indikasi bahwa seseorang dapat melanjutkan dan bekromunikasi. Hal yang sama berlaku di jaringan dibutuhkan dua (atau lebih) entitas berkomunikasi yang menjalankan protokol yang sama untuk menyelesaikan tugas.

› Protokol Jaringan

Protokol jaringan mirip dengan protokol manusia, kecuali bahwa entitas yang bertukar pesan dan mengambil tindakan adalah komponen perangkat keras atau perangkat lunak dari beberapa perangkat (misalnya, komputer, smartphone, tablet, router, atau perangkat yang mendukung jaringan lainnya). Semua aktivitas di internet itu melibatkan dua atau lebih banyak entitas remote yang berkomunikasi diatur oleh protokol.

B. Tepi Jaringan

Kita mulai di bagian ini pada tepi jaringan dan melihat komponen yang paling sering kita jumpai yaitu, komputer, smartphone, dan perangkat lain yang kita gunakan setiap hari. Komputer dan perangkat lainnya tersebut terhubung ke Internet sering disebut sebagai sistem akhir karena mereka duduk di tepi Internet. Sistem akhir juga disebut sebagai host karena mereka meng-host (yaitu, menjalankan) program aplikasi.

1. Mengakses Jaringan

Saat ini, dua jenis akses broadband perumahan yang paling umum adalah **Digital Subscriber Line (DSL)** dan kabel. perumahan

biasanya memperoleh akses Internet DSL dari perusahaan telepon lokal yang sama (Telco) yang menyediakan akses telepon lokal. Jadi, ketika DSL digunakan, telekomunikasi pelanggan juga miliknya ISP. Modem DSL setiap pelanggan menggunakan saluran telepon yang ada untuk bertukar data dengan digital subscriber line access multiplexer (DSLAM) yang terletak di kantor pusat lokal Telco. Modem DSL rumah mengambil data digital dan menerjemahkannya ke nada frekuensi tinggi untuk pengiriman melalui kabel telepon ke kantor pusat. Sinyal analog dari banyak rumah semacam itu diterjemahkan kembali ke dalam format digital di DSLAM.

2. Media Fisik

Saat mengirim dari sumber ke tujuan, melewati serangkaian pasangan transmitter-receiver. Untuk setiap pasangan transmitter-receiver, bit dikirim dengan menyebarkan gelombang elektromagnetik atau pulsa optik yang melintasi media fisik. Contoh media fisik termasuk twisted-pair kawat tembaga, kabel koaksial, kabel serat optik multimode, spektrum radio terestrial, dan radio satelit spektrum.

Media fisik terbagi dalam dua kategori: **guided media** dan **unguided media**. Dengan guided media, gelombang dipandu sepanjang media padat, seperti kabel serat optik, kawat tembaga twisted-pair, atau kabel koaksial. Dan unguided media, gelombang merambat di atmosfer dan di luar ruang, seperti dalam LAN nirkabel atau saluran satelit digital.

C. Inti jaringan

1. Paket Switching

Dalam aplikasi jaringan, sistem akhir menukar pesan satu sama lain. Pesan dapat berisi apa pun yang diinginkan oleh perancang aplikasi. Untuk mengirim pesan dari sistem ujung sumber ke sistem akhir tujuan, sumber memecah pesan panjang menjadi potongan data kecil yang dikenal sebagai **paket**. Antara sumber dan tujuan, setiap paket melakukan perjalanan melalui tautan komunikasi dan **sakelar paket** (dua jenis yang utama yaitu; **router** dan **link-layer switch**). Paket ditransmisikan setiap tautan komunikasi dengan kecepatan yang sama dengan laju transmisi penuh dari tautan tersebut.

2. Circuit Switching

Ada dua pendekatan mendasar untuk memindahkan data melalui tautan jaringan dan sakelar: **Sircuit Switching** dan **Packet Switching.** Dalam jaringan circuit-switched, sumber daya dibutuhkan sepanjang jalur (buffer, link transmission rate) untuk menyediakan komunikasi antara sistem akhir yang dicadangkan untuk durasi sesi komunikasi antara sistem akhir. Dalam jaringan packet-switched, sumber daya ini tidak dicadangkan; Sebuah pesan sesi menggunakan sumber daya sesuai permintaan dan sebagai konsekuensinya, mungkin harus menunggu (yaitu, antrian) untuk akses ke tautan komunikasi.

3. A Network Of Network

Kita melihat sebelumnya bahwa sistem akhir (PC, smartphone, server Web, server mail, dan sebagainya) terhubung ke Internet melalui akses ISP. Akses ISP dapat menyediakan konektivitas kabel atau nirkabel, menggunakan berbagai akses teknologi termasuk DSL, kabel, FTTH, Wi-Fi, dan seluler. Satu pendekatan naif adalah memiliki setiap akses ISP secara langsung terhubung dengan setiap akses ISP lainnya. Desain mesh seperti itu, tentu saja, terlalu mahal untuk itu akses ISP, karena akan memerlukan setiap akses ISP untuk memiliki tautan komunikasi terpisah untuk masing-masing ratusan ribu ISP akses lainnya di seluruh dunia.

D. Keterlambatan, Kehilangan, dan Throughput dalam Packet-Switched Networks

1. Gambaran Umum Keterlambatan dalam Jaringan Packet-Switched

Suatu paket dimulai pada sebuah host (sumber), melewati serangkaian router, dan mengakhiri perjalanan di host lain (tujuan). Perjalanan paket dari satu node (host atau router) ke node berikutnya (host atau router) di sepanjang jalur ini, paket tersebut mengalami beberapa jenis keterlambatan di masing-masing node di sepanjang jalan.

2. Keterlambatan Antrian dan Kehilangan Paket

Antrian memiliki kapasitas yang terbatas, meskipun kapasitas antriannya sangat besar tergantung pada desain dan biaya router. Karena kapasitas antrian terbatas, penundaan paket tidak terlalu

mendekati tak terhingga ketika intensitas lalu lintas mendekati 1. Sebagai gantinya, sebuah paket dapat tiba untuk menemukan antrian penuh. Dengan tidak ada tempat untuk menyimpan paket tersebut, router akan melakukannya penurunan paket itu; artinya, paket itu akan hilang.

3. **Keterlambatan Ujung ke Ujung (End to End Delay)**

   Untuk memahami konsep ini, anggaplah ada adalah router N-1 antara host sumber dan host tujuan. kita anggap saja untuk saat ini bahwa jaringan tidak kebobolan (sehingga penundaan antrian dapat diabaikan), penundaan pemrosesan di masing-masing router dan pada host sumber adalah dproc, laju transmisi keluar dari masing-masing router dan keluar dari host sumber adalah R bit / detik, dan propagasi pada setiap tautan adalah dproc. Penundaan nodal terakumulasi dan memberikan end-to-end delay.

   $$d_{end-end}=N(d_{proc}+d_{trans}+d_{prop})$$

   Di mana sekali lagi dtrans=L/R, di mana L adalah ukuran paket.

4. **Hasil Dalam Jaringan Komputer**

   Selain penundaan dan kehilangan paket, ukuran kinerja penting lainnya dalam jaringan komputer adalah end-to-end throughput. Untuk beberapa aplikasi, seperti Internet telepon, diharapkan untuk memiliki penundaan yang rendah dan throughput instan secara konsisten di atas beberapa ambang batas (misalnya, lebih dari 24 kbps untuk beberapa aplikasi telepon Internet dan lebih dari 256 kbps untuk beberapa aplikasi video waktu nyata). Untuk aplikasi lain, termasuk yang melibatkan transfer file, diharapkan untuk memiliki throughput setinggi mungkin.

E. Lapisan Protokol dan Model Layanannya

- **Arsitektur Layer**

  a. *Layering Protocol*

     Untuk memberikan struktur kepada desain protokol jaringan, perancang jaringan mengatur protokol perangkat keras jaringan dan perangkat lunak yang mengimplementasikan protokol dalam lapisan.

b. *Aplication Layer*

Lapisan aplikasi adalah tempat aplikasi jaringan dan protokol lapisan aplikasi berada. Lapisan aplikasi Internet mencakup banyak protokol, seperti protokol HTTP (yang menyediakan untuk permintaan dokumen web dan transfer), SMTP (yang menyediakan untuk transfer pesan email), dan FTP (yang menyediakan untuk transfer file antara dua sistem akhir).

c. *Transport Layer*

Lapisan transport Internet mengangkut pesan layer aplikasi antara aplikasi end-point. Di Internet ada dua protokol transport, TCP dan UDP, yang keduanya dapat mengangkut pesan layer aplikasi. TCP menyediakan koneksi-layanan yang berorientasi pada aplikasinya. Layanan ini termasuk jaminan pengiriman pesan lapisan aplikasi ke tujuan dan kontrol aliran (yaitu,pencocokan kecepatan pengirim / penerima). TCP juga memecah pesan panjang menjadi segmen yang lebih pendek dan menyediakan mekanisme kontrol kemacetan, sehingga sumber menahan laju transmisinya ketika jaringan sedang padat. Protokol UDP menyediakan layanan tanpa koneksi ke aplikasinya. Ini tanpa embel-embel layanan yang tidak menyediakan keandalan, kontrol aliran, dan kontrol kemacetan.

d. *Network Layer*

Lapisan jaringan Internet bertanggung jawab untuk memindahkan paket lapisan jaringan yang dikenal sebagai **datagram** dari satu host ke yang lain. Protokol lapisan transport Internet (TCP atau UDP) di host sumber melewati segmen transport-layer dan alamat tujuan ke lapisan jaringan. Lapisan jaringan kemudian menyediakan layanan mengirimkan segmen ke lapisan transportasi di host tujuan. Lapisan jaringan Internet termasuk protokol IP, yang mendefinisikan bidang dalam datagram serta bagaimana sistem akhir dan router bekerja pada bidang ini. Hanya ada satu protokol IP, dan semuan Komponen internet yang memiliki lapisan jaringan harus menjalankan protokol IP. Lapisan jaringan Internet juga berisi protokol perutean yang menentukan rute yang diambil datagram antara sumber dan tujuan.

e. *Link-Layer*

Lapisan jaringan Internet merutekan datagram melalui serangkaian router antara sumber dan tujuan. Untuk memindahkan paket dari satu node (host atau router) ke node berikutnya dalam rut, jaringan layer bergantung pada layanan dari layer tautan. Secara khusus, pada setiap node, lapisan jaringan melewati datagram ke lapisan tautan, yang mengirimkan datagram ke simpul berikutnya di sepanjang rute. Pada node berikutnya, lapisan tautan meneruskan datagram ke lapisan jaringan. Layanan yang diberikan oleh lapisan tautan bergantung pada protokol lapisan tautan khusus yang digunakan tautannya.

f. Lapisan fisik

Sedangkan tugas dari layer link adalah memindahkan seluruh frame dari satu elemen jaringan ke elemen jaringan yang berdekatan, tugas dari lapisan fisik adalah untuk memindahkan bit individu dalam bingkai dari satu node ke node berikutnya. Protokol dalam lapisan ini sekali lagi bergantung pada tautan dan selanjutnya bergantung pada transmisi yang sebenarnya media tautan (misalnya, kawat tembaga twisted-pair, serat optik mode tunggal).

g. *Model OSI*

Model OSI terbentuk ketika protokol yang akan menjadi protokol Internet yang masih baru, dan hanyalah salah satu dari banyak suite protokol yang berbeda dalam pengembangan. Tujuh lapisan model referensi OSI adalah: lapisan aplikasi, lapisan presentasi, lapisan sesi, lapisan transportasi, lapisan jaringan, lapisan tautan data, dan lapisan fisik.

BAB 2
APPLICATION LAYER

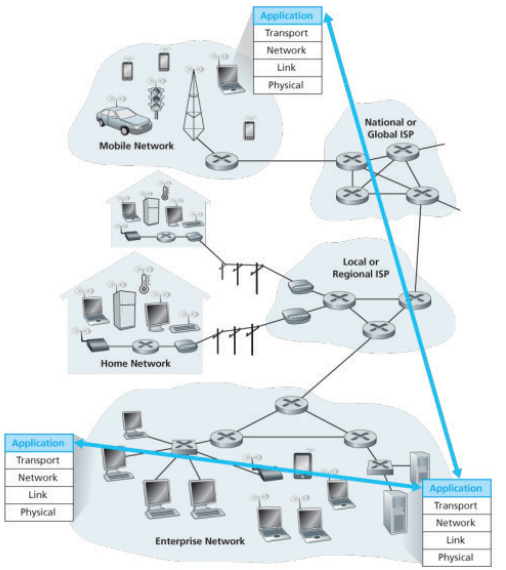
Lapisan aplikasi adalah suatu teknologi yang digunakan untuk mengelompokkan protokol dan metode dalam model arsitektur jaringan komputer. Baik model OSI maupun TCP/IP memiliki suatu lapisan aplikasi. Singkatnya, application layer merupakan lapisan OSI Layer yang menyediakan interface atau antar muka antar aplikasi yang digunakan untuk melakukan komunikasi di dalam jaringan, dan kemudian membantu mengerimkan dan menerima pesan yang dikirimkan di dalam jaringan tersebut. Ini juga di namakan protokol jaringan yang mengatur tugas-tugas tertentu dalam suatu jaringan internet. Aplikasi internet termasuk aplikasi berbasis teks klasik yang menjadi populer di tahun 1970-an.

Selama periode saat ini semakin berkembang, kita telah melihat munculnya generasi baru aplikasi jejaring sosial-seperti Facebook, Instagram, Twitter, dan WeChat sehingga kita bisa terhubung satu sama lain.

Agar dapat mencapai tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan layanan (service). Pihak yang meminta layanan disebut klien (client) dan yang memberikan layanan disebut pelayan (server). Arsitektur ini disebut dengan sistem client-server, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

A. Prinsip Jaringan Aplikasi

1. Arsitektur Jaringan Aplikasi



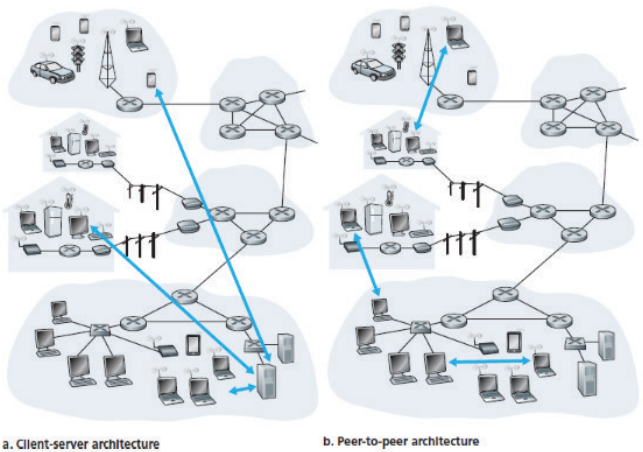
Gambar 2.1 Arsitektur Jaringan Aplikasi

Arsitektur jaringan komputer merupakan tata cara penggunaan perangkat keras dan perangkat lunak dalam jaringan agar satu komputer dengan komputer lainnya dapat melakukan komunikasi dan pertukaran data.

Keamanan sistem jaringan komputer adalah bagian tak terpisahkan dari keamanan sistem komputer sebuah organisasi secara keseluruhan, terutama dengan semakin berkembangnya Internet. Semakin banyak aplikasi pengguna yang berbasiskan pada jaringan komputer. Jika sebuah jaringan komputer tidak aman, maka sistem komputer pada organisasi tersebut juga tidak aman.

Dalam arsitektur klien-server, selalu ada host, yang disebut server, yang diminta oleh layanan dari banyak host lain, yang disebut klien. Karakteristik lain dari Arsitektur klien-server adalah bahwa server memiliki alamat tetap yang terkenal, yang disebut alamat IP (yang akan segera kita diskusikan). Karena server memiliki alamat tetap, terkenal, dan karena server selalu aktif, klien selalu dapat menghubungi server dengan mengirimkan paket ke alamat IP server.

Arsitektur sistem P2P Peer-to-peer sering menerapkan sistem jaringan overlay abstrak, dibangun di Application Layer, di atas topologi jaringan asli atau fisik. Lapisan tersebut digunakan untuk penemuan pengindeksan dan peer dan membuat sistem P2P independen dari topologi jaringan fisik. Konten biasanya dipertukarkan langsung melalui jaringan Internet Protocol yang mendasari (IP). Pada jaringan peer-to-peer terstruktur, kadang-kadang, sumber daya diatur dengan kriteria dan algoritma khusus, yang menyebabkan lapisan dengan topologi lebih spesifik.



Gambar 2.2 Client Server dan Peer to Peer Arsitektur

2. Proses Berkomunikasi

Sebelum membangun aplikasi jaringan, Anda juga memerlukan pemahaman dasar tentang program, supaya berjalan baik dalam mengaplikasikan jaringan, berkomunikasi satu sama lain dalam perangkat jaringan supaya nyambung dan paham jalanya program seperti apa dalam jaringan sistem operasi. Sebuah proses dapat dianggap sebagai sebuah program yang berjalan dalam sistem akhir, ketika proses berjalan pada sistem akhir yang sama, mereka dapat berkomunikasi satu sama lain dengan komunikasi antarproses, menggunakan aturan yang diatur oleh sistem operasi.

› Proses Client dan Server

Client adalah sebuah sistem atau proses yang melakukan permintaan data atau layanan ke server. Sedangkan server ialah,

sistem atau proses yang menyediakan data atau layanan yang diminta oleh client. Cara kerja jaringan client dan server adalah komputer yang bertindak sebagai client akan meminta data dari komputer server dan komputer server akan melayani permintaan data dari komputer client, client dan server terhubung pada suatu jaringan komputer. Untuk spesifikasi komputer, komputer server memiliki spesifikasi yang lebih tinggi daripada komputer client.

› Proses Pengalamatan

Seperti contoh untuk mengirim surat pos ke tujuan tertentu, tujuan perlu memiliki alamat. Demikian pula agar proses yang berjalan pada satu host untuk mengirim paket tujuan yang berjalan di lain host, tujuan penerimaan harus memiliki alamat dengan jelas. Untuk mengidentifikasi proses penerimaan, dua lembar informasi perlu ditentukan: (1) alamat host dan (2) pengidentifikasi yang menentukan proses penerimaan di host tujuan.

3. Layanan Transportasi Tersedia untuk Aplikasi

Ingatlah bahwa soket adalah antarmuka antara proses aplikasi dan protokol transport-layer. Aplikasi di sisi pengiriman mendorong pesan melalui soket. Di sisi lain protokol transport-layer memiliki tanggung jawab mendapatkan pesan ke soket proses penerimaan.

Protokol adalah bagian yang penting dalam proses pertukaran informasi antar komputer yang mengatur proses pertukaran data antar komputer. Teknologi protokol dapat diterapkan pada perangkat lunak, perangkat keras atau kombinasi dari keduanya. Protokol sangat berhubungan dengan teknologi informasi dan tidak bisa lepas dari aktivitas di internet.

Sebagai contoh, seperti orang yang mengirimkan email. Email dalam komputer bisa disebut dengan sebuah data. Sehingga email yang dikirimkan pada seseorang dari komputer satu ke komputer lain sebenarnya adalah pengiriman data. Setiap orang mengirimkan email, pasti email akan melewati beberapa protokol. Semua protokol harus dilalui agar email bisa keluar dan diterima komputer lain pada jaringan yang sama atau berbeda.

Bandwidth adalah ukuran dari banyaknya informasi yang dapat dialirkan dari source ke destination dalam waktu tertentu. Bandwidth

dapat dipakai untuk mengukur aliran data analog atau aliran data digital. Secara umum pelangan internet, informasi yang digunakna adalah jenis data digital. Satuan yang digunakan untuk bandwdth digital adalah bps (bit per second).

Throughput adalah bandwidth yang aktual atau sebenarnya, yang diukur dengan satuan waktu tertentu dan pada kondisi jaringan tertentu yang digunakan untuk melakukan transfer data dengan ukuran tertentu pula.

Keamanan jaringan atau yang biasa disebut sebagai Network Security biasanya dilakukan untuk mencegah dan memantau akses yang tidak sah, penyalahgunaan, modifikasi, dan lain-lain. Dimana tindakan pencegahan untuk melindungi jaringan tersebut merupakan tugas dari seorang administrator jaringan.

4. Layanan transportasi yang disediakan oleh Internet

Ketika sebuah komputer dikoneksikan dengan komputer lainnya, mereka akan berkomunikasi lewat protokol. Protokol yang paling terkenal adalah protokol TCP/IP (Transmission Control Protocol/ Internet Protocol). Pengertian TCP/IP tak bisa lepas dari fakta bahwa ada dua jenis protokol yang digunakan pada jaringan ini, yaitu protokol TCP dan protokol IP. TCP/IP adalah suatu standar komunikasi yang dapat digunakan untuk bertukar data antar komputer oleh suatu komunitas yang tergabung melalui jaringan internet.

Dari pengertian TCP/IP tadi bisa kita pahami bahwa :

- Ada lebih dari 1 komputer yang tergabung dan berkomunikasi menggunakan TCP/IP.
- Komputer yang terkoneksi melalui protokol TCP/IP melakukan sharing data.
- Komputer yang terkoneksi TCP/IP juga berarti terkoneksi dengan "internet".

UDP adalah jenis protokol internet yang memungkinkan sebuah perangkat lunak pada komputer bisa mengirimkan pesan ke komputer lain melalui jaringan tanpa perlu ada komunikasi awal. Karakteristik UDP merupakan jenis protokol yang memiliki karakteristik connectionless atau tidak berbasis koneksi. Aplikasi UDP adalah SunRPC, SNMP, DNS, dan TFTP. Lalu contoh aplikasi

untuk TCP antara lain FTP, SMTP, dan TELNET. Jenis Port yang Digunakan UDP menggunakan port 16 bit integer yang dibagi menjadi tiga bagian, yakni 49152-65535 untuk ephemeral port, port 1-1023 untuk well-known port, dan port 1024-49151 untuk registered port. Proses transmisi data UDP dilakukan dalam bentuk datagram yang memungkinkan data yang diterima bisa mengalami kerusakan dan tidak urut. Berbeda dengan UDP, TCP memiliki dua jalur yang digunakan untuk melakukan pertukaran data yang masuk dan keluar.

5. **Protokol aplikasi-Layer**

   Contoh Protokol-Protokol di Aplication Layer pada Lapisan OSI

   a. DHCP (Dynamic Host Configuration Protocol)
   b. Domain Name System (DNS)
   c. HTTP (Hypertext Transfer Protocol http)
   d. FTP (File Transfer Protocol)
   e. Telnet
   f. Simple Mail Transfer Protocol (SMTP)
   g. Simple Network Management Protocol (SNMP)
   h. Network File System (NFS)

B. Web dan HTTP

Hypertext Transfer Protocol (HTTP) adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Penggunaannya menggunakan sumber daya yang saling terhubung dengan tautan, yang disebut dengan dokumen hiperteks, yang kemudian membentuk World Wide Web pada tahun 1990 oleh fisikawan Inggris, Tim Berners-Lee. Hingga kini, ada dua versi mayor dari protokol HTTP, yakni HTTP/1.0 yang menggunakan koneksi terpisah untuk setiap dokumen, dan HTTP/1.1 yang dapat menggunakan koneksi bersama untuk melakukan transaksi. Dengan demikian, HTTP/1.1 bisa lebih cepat karena memang tidak usah membuang waktu untuk pembuatan koneksi berulang-ulang.

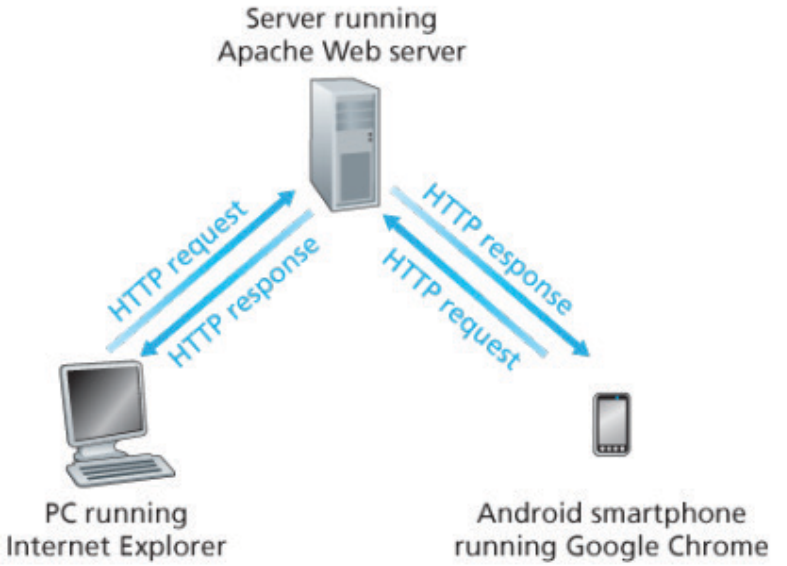
Pengertian URL (uniform resource locator) adalah rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di

Internet. URL pertama kali diciptakan oleh Tim Berners-Lee pada tahun 1991 agar penulis-penulis dokumen dokumen dapat mereferensikan pranala ke World Wide Web. Sejak 1994, konsep URL telah dikembangkan menjadi istilah Uniform Resource Identifier (URI) yang lebih umum sifatnya.

1. Gambaran HTTP

HyperText Transfer Protocol (HTTP) pada Web Application-layer Protocol, adalah inti dari Web. Hal ini didefinisikan dalam [RFC 1945] dan [RFC 2616]. HTTP diimplementasikan dalam dua program: klien program dan program server. Program klien dan server program, mengeksekusi pada akhir yang berbeda berbicara satu sama lain dengan bertukar pesan HTTP.

Sebuah halaman web (juga disebut dokumen) terdiri dari objek. Sebuah objek hanyalah sebuah file seperti sebuah HTML file, gambar JPEG, applet Java, atau klip video yang dapat diakses oleh satu URL. Kebanyakan web Halaman terdiri dari file HTML dasar dan beberapa objek yang direferensikan. Misalnya, jika halaman web berisi teks HTML dan lima gambar JPEG, maka halaman web memiliki enam objek: file HTML dasar ditambah lima gambar. File HTML dasar merujuk pada objek lain di halaman dengan URL objek. Setiap URL memiliki dua komponen: nama host server yang memiliki objek dan jalur objek Nama. Sebagai contoh, URL (http://www.someschool.edu/someDepartment/picture.gif). memiliki www.someSchool.edu untuk hostname dan/someDepartment/picture.gif untuk jalur Nama. Karena web browser (seperti Internet Explorer dan Firefox) menerapkan sisi klien HTTP dalam konteks web, kita akan menggunakan kata browser dan klien secara bergantian. Web Server, yang mengimplementasikan sisi server HTTP, masing-masing dialamatkan oleh URL.



Gambar 2.3 http Protocol

HTTP memiliki fungsi koneksi persisten yang memungkinkan saluran tetap terbuka alih-alih ditutup setelah pertukaran data yang diminta. TCP memulai koneksi setelah konfirmasi dari kedua ujungnya bahwa mereka tersedia dan terbuka untuk pertukaran data. Dalam koneksi non-persisten, saluran akan ditutup ketika satu host memberi sinyal bahwa ia ingin mengakhiri komunikasi atau ketika sejumlah waktu telah berlalu tanpa pertukaran data. Untuk menjaga koneksi yang persisten, paket TCP keep-live dikirim untuk mencegah koneksi dari waktu habis.

Sebuah HTTP "klien" adalah sebuah aplikasi (browser Web atau klien lainnya), dengan mengirimkan permintaan untuk terhubung ke server untuk mencapai satu atau lebih dari tujuan HTTP server. Sebuah HTTP "server" juga merupakan aplikasi (biasanya layanan Web, seperti Apache Web Server atau server IIS, dll), dengan menerima permintaan klien dan mengirim data respon HTTP.

2. Format Pesan HTTP

Di bawah ini pesan permintaan HTTP:

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

Gambar 2.4 Pesan Permintaan HTTP

Respon HTTP khas pesan

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 18 Aug 2015 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 18 Aug 2015 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data ...)
```

Gambar 2.5 Response HTTP

Pertama Telnet ke server web favorit Anda. Kemudian ketik pesan permintaan satu baris untuk beberapa objek yang disimpan di server. Sebagai contoh, jika Anda memiliki *command prompt*, ketik:

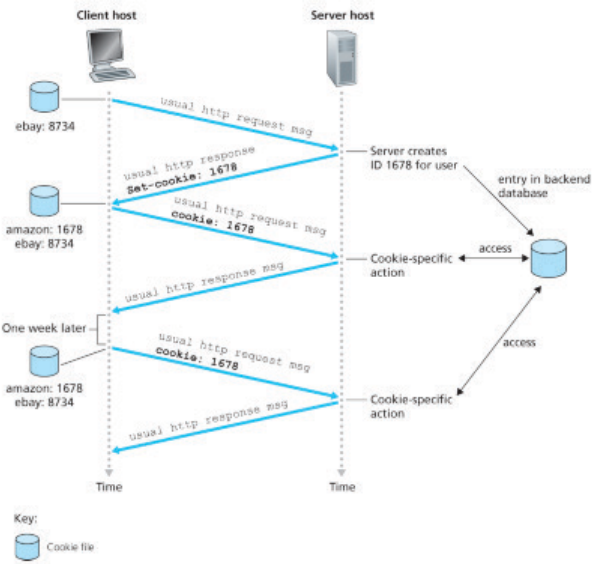
```
telnet gaia.cs.umass.edu 80

GET /kurose_ross/interactive/index.php HTTP/1.1
Host: gaia.cs.umass.edu
```

Gambar 2.6 Telnet ke Server

3. Interaksi Pengguna-Server: Cookies

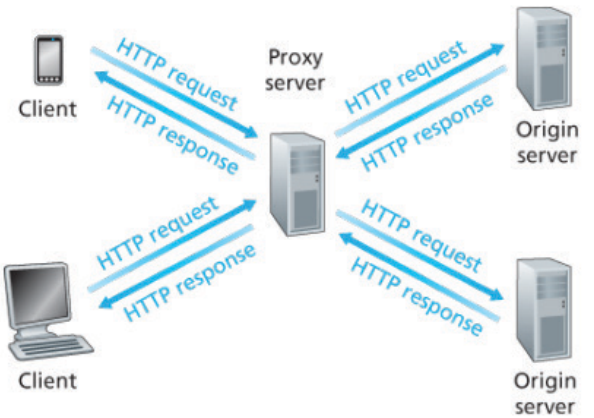
Server HTTP tidak memiliki kewarganegaraan, sehingga ini menyederhanakan desain server web yang kinerjanya tinggi yang dapat menangani ribuan TCP secara bersamaan koneksi. Sering kali situs web diinginkan untuk mengidentifikasi pengguna, baik karena server ingin membatasi akses pengguna atau karena ingin menyajikan konten sebagai identitas utama pengguna. Untuk tujuan ini, HTTP

menggunakan cookie. Cookie didefinisikan dalam [RFC 6265], memungkinkan situs untuk melacak pengguna.



Gambar 2.7 Teknologi Cookie

Teknologi cookie memiliki empat komponen: (1) baris header cookie di HTTP pesan tanggapan; (2) baris header cookie di pesan permintaan HTTP; (3) file cookie disimpan di sistem akhir pengguna dan dikelola oleh browser pengguna; dan (4) database back-end di situs Web.



Gambar 2.8 Komponen Teknologi Cookie

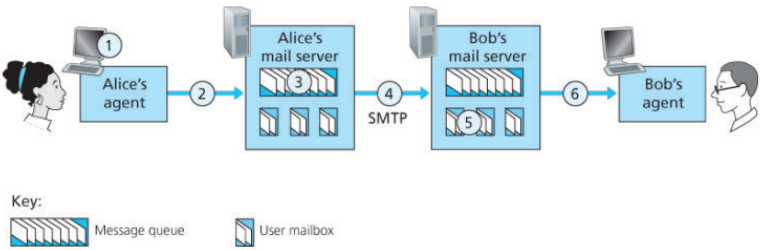
4. Web Caching

Cache adalah proses penyimpanan sementara data atau halaman HTML dan gambar sebuah website untuk mengurangi penggunaan bandwidth dan loading server. Secara sederhana, cache adalah teknologi yang membantu menampilkan halaman website lebih cepat. Berbeda dengan cookies yang merekam jejak dan aktivitas pengguna ketika berselancar di internet. Proses permintaan data sebelumnya yang sudah di cari maka akan tersimpan dalam server dan tidak usah reques lagi ke internet dalam lebih cepat dalam mengirim data.

C. Pesan Elektronik dan Internet

1. SMTP

Simple Mail Transfer Protocol atau SMTP adalah suatu protokol untuk berkomunikasi dengan server guna mengirimkan email dari lokal email ke server, sebelum akhirnya dikirimkan ke server email penerima. Proses ini dikontrol dengan Mail Transfer Agent (MTA) yang ada dalam server email Anda.



Gambar 2.9 SMTP

Selanjutnya Mari kita lihat contoh transkrip pesan yang dipertukarkan antara klien SMTP (C) dan server SMTP. Nama host klien adalah crepes.fr dan nama host server hamburger.edu. Garis teks ASCII diawali dengan C: persis dengan baris yang dikirim klien ke Soket TCP, dan garis teks ASCII diawali dengan S: persis baris server mengirimkan ke Soket TCP. Transkrip berikut dimulai segera setelah sambungan TCP dibuat.

```
S:  220 hamburger.edu
C:  HELO crepes.fr
S:  250 Hello crepes.fr, pleased to meet you
C:  MAIL FROM: <alice@crepes.fr>
S:  250 alice@crepes.fr ... Sender ok
C:  RCPT TO: <bob@hamburger.edu>
S:  250 bob@hamburger.edu ... Recipient ok
C:  DATA
S:  354 Enter mail, end with "." on a line by itself
C:  Do you like ketchup?
C:  How about pickles?
C:  .
S:  250 Message accepted for delivery
C:  QUIT
S:  221 hamburger.edu closing connection
```

Gambar 2.10 Contoh Transkip Pesan Klien SMTP

Perbedaan kedua, yang kita singgung sebelumnya, adalah bahwa SMTP memerlukan setiap pesan, termasuk pesan untuk berada dalam format ASCII 7-bit. Jika pesan berisi karakter yang tidak 7-bit ASCII (misalnya, karakter Perancis dengan aksen) atau berisi data biner (seperti file gambar), maka pesan harus dikodekan ke dalam ASCII 7-bit. HTTP data.

2. **Format pesan email.**

   Jika Anda belum memiliki alamat email, Anda harus mendaftar di penyedia layanan email sebelum melanjutkan. Untungnya, banyak layanan email berbasis web gratis yang memungkinkan Anda mendapat alamat email tanpa mengeluarkan uang. Beberapa layanan email web paling populer di antaranya Gmail.

   Header pesan tipikal terlihat seperti ini:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the meaning of life.
```

Gambar 2.11 Format Pesan SMTP

3. Protokol Akses Surat

› POP3

POP3 (Post Office Protocol version 3) digunakan untuk berkomunikasi dengan server email remote dan men-download semua email ke dalam aplikasi email client seperti Outlook, Thunderbird, Windows Mail, Mac Mail, dll. Biasanya, aplikasi email client memiliki opsi untuk meninggalkan salinan email yang telah di download tetap berada di server atau tidak. Jika Anda mengakses akun email yang sama dari perangkat yang berbeda, sangat disarankan untuk tetap meninggalkan salinan email di server. Bila tidak, maka perangkat Anda yang lain tidak akan bisa men-download email apapun jika perangkat pertama telah menghapus email-email tersebut dari server (melalui fitur POP3). Kita bisa juga menyebut POP3 ini sebagai protokol komunikasi 1 arah, artinya data akan ditarik dari server remote dan dikirim langsung ke client. Fase otorisasi memiliki dua perintah utama: user <username> dan Pass <password>. Untuk mengilustrasikan kedua perintah ini, kami menyarankan bahwa Telnet langsung ke POP3 server, menggunakan Port 110, dan mengeluarkan perintah ini. Misalkan mailServer adalah nama server email.

```
telnet mailServer 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

Gambar 2.12 Implementasi POP3

› IMAP

IMAP (Internet Message Access Protocol) seperti POP3 juga digunakan untuk mengambil email ke aplikasi email client, namun, IMAP memiliki perbedaan yang cukup besar – karena hanya informasi header email saja yang akan di-download, sedangkan email yang asli tetap akan ditinggalkan di server. Ini berbeda dengan POP3 yang justru memindahkan semua email ke aplikasi email client dan tidak menyisakan email di server. IMAP tergolong sebagai protokol komunikasi 2 arah, karena

perubahan yang dibuat di aplikasi email client akan dikirimkan juga ke server. Sehingga pada akhirnya, protokol ini menjadi lebih populer karena penyedia layanan email seperti GMail, dll, justru merekomendasikan untuk menggunakan IMAP daripada POP3.

Port default IMAP adalah:

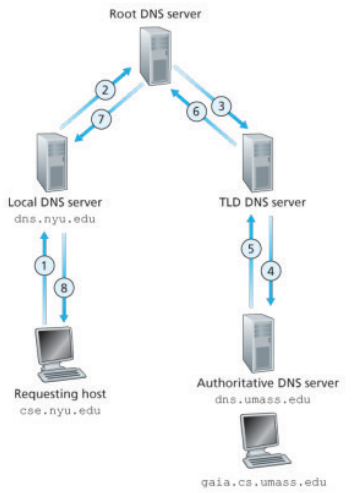
- Port 143 – port tidak terenkripsi (non-encrypted)
- Port 993 – port SSL/TLS, yang juga dikenal sebagai IMAPS

D. DNS – Layanan Direktori Internet

Domain Name System atau DNS adalah sebuah sistem yang memungkinkan manusia dan komputer untuk berkomunikasi secara lebih mudah. Manusia menggunakan nama, komputer menggunakan angka, dan DNS berada di antara mereka untuk menyesuaikan nama dengan angka dalam daftar tertentu. Kita bisa mengambil contoh aplikasi Kontak pada ponsel pintar atau smartphone. Cara kerja DNS meliputi beberapa langkah dan melalui struktur DNS. Langkah pertama dimulai dengan sebuah DNS query, sebuah permintaan informasi. Awalnya, server DNS akan mencari informasi di dalam filehost sebuah file plain text dari sistem operasi yang bertanggung jawab atas pemetaan hostname ke alamat IP. Jika tidak ada informasi yang ditemukan, server akan mencari cache sebuah komponen hardware atau software yang menyimpan data untuk sementara.

- **Caching DNS**

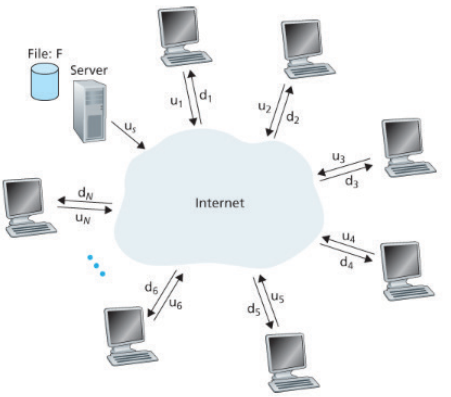
  DNS Cache atau kadang bisa juga disebut DNS Resolver Cache merupakan database sementara pada sistem operasi komputer yang menyimpan rekaman data IP dari nama domain yang sebelumnya telah dikunjungi oleh pengguna komputer.



Gambar 2.13 Ilustrasi Caching DNS

E. Distribusi File Peer-to-Peer

Aplikasi P2P yang sangat alami, yaitu mendistribusikan file besar ke sejumlah besar host (disebut rekan). Dalam distribusi file klien-server, server harus mengirim salinan file ke masing-masing rekan – menempatkan beban yang sangat besar pada server dan menghabiskan banyak bandwidth server. Dalam distribusi file P2P, setiap rekan dapat mendistribusikan kembali bagian mana pun dari file yang telah diterimanya kepada yang lain rekan kerja, sehingga membantu server dalam proses distribusi.



Gambar 2.14 Ilustrasi Distribusi File Peer to Peer

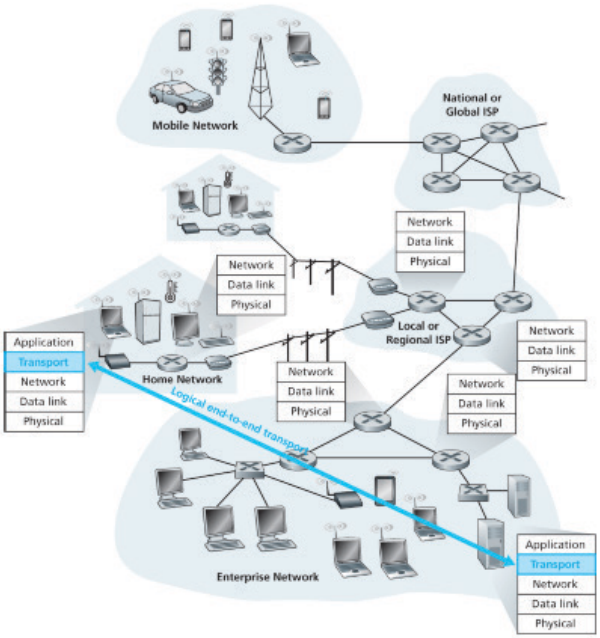
BAB 3
LAPISAN TRANSPORT

Berada di antara lapisan aplikasi dan jaringan, lapisan transport adalah bagian utama dari arsitektur jaringan berlapis. Ini memiliki peran penting dalam menyediakan layanan komunikasi langsung ke proses aplikasi yang berjalan pada host yang berbeda. Penekanan khusus akan diberikan pada protokol Internet, khususnya protokol transport-layer TCP dan UDP.

A. Pengantar dan Layanan Lapisan Transport

Protokol *transport-layer* menyediakan komunikasi logis antara proses aplikasi yang berjalan pada host. Dengan komunikasi logis, maksud kita adalah dari perspektif aplikasi, seolah-olah host yang menjalankan proses terhubung langsung; pada kenyataannya, host mungkin berada di sisi berlawanan, terhubung melalui banyak router dan berbagai jenis tautan. Proses aplikasi menggunakan komunikasi logis yang disediakan oleh lapisan transportasi untuk saling mengirim pesan, bebas dari kekhawatiran akan detail infrastruktur fisik yang digunakan untuk membawa pesan-pesan ini.

1. Hubungan Antara Transportasi dan Lapisan Jaringan



Gambar 3.1 Relasi Transportasi dan Lapisan Jaringan

Lapisan transport terletak tepat di atas lapisan jaringan dalam tumpukan protokol. Sedangkan protokol transport-layer menyediakan komunikasi logis antara proses berjalan pada host yang berbeda, protokol lapisan jaringan menyediakan komunikasi logis antar host. Perbedaan ini halus tetapi penting. Layanan POS memindahkan surat dari rumah ke rumah, bukan dari orang ke orang.

Protokol transport-layer hidup di sistem akhir. Dalam sistem akhir, protokol transport memindahkan pesan dari proses aplikasi ke tepi jaringan (yaitu, lapisan jaringan) dan sebaliknya, tetapi tidak ada yang mengatakan tentang bagaimana pesan dipindahkan dalam inti jaringan. Bahkan, router perantara tidak bekerja, atau mengenali, informasi apa pun yang mungkin ditambahkan oleh layer transport ke pesan aplikasi.

Namun demikian, layanan tertentu dapat ditawarkan oleh protokol transportasi bahkan ketika protokol jaringan yang mendasarinya tidak menawarkan layanan yang sesuai pada lapisan jaringan. Misalnya, seperti yang akan kita lihat di bab ini, protokol transport

dapat menawarkan layanan transfer data yang dapat diandalkan ke suatu aplikasi bahkan ketika protokol jaringan yang mendasarinya tidak dapat diandalkan, bahkan ketika protokol jaringan kehilangan, kerusakan, atau duplikat paket. Sebagai contoh lain, protokol transport dapat menggunakan enkripsi untuk menjamin bahwa pesan aplikasi tidak dibaca oleh penyusup, bahkan ketika lapisan jaringan tidak dapat menjamin kerahasiaan segmen lapisan transport.

2. Gambaran Umum tentang Transport Layer di Internet

Ingatlah bahwa Internet membuat dua protokol transport-layer yang berbeda tersedia untuk lapisan aplikasi. Salah satu protokol ini adalah UDP (*User Datagram Protocol*), yang menyediakan layanan tanpa koneksi yang tidak dapat diandalkan untuk aplikasi yang memohon. Protokol kedua adalah TCP (*Transmission Control Protocol*), yang menyediakan layanan yang dapat diandalkan, berorientasi koneksi ke aplikasi yang memohon. Saat merancang aplikasi jaringan, pengembang aplikasi harus menentukan salah satu dari dua protokol transport ini.

Untuk menyederhanakan terminologi, kita merujuk ke paket transport-layer sebagai segmen. Kita menyebutkan, bagaimanapun, bahwa literatur Internet (misalnya, RFC) juga mengacu pada paket transport-layer untuk TCP sebagai segmen tetapi sering merujuk pada paket untuk UDP sebagai datagram. Tetapi literatur Internet yang sama ini juga menggunakan istilah datagram untuk paket lapisan jaringan.

Setelah melihat sekilas pada model layanan IP, mari sekarang meringkas model layanan yang disediakan oleh UDP dan TCP. Tanggung jawab paling mendasar dari UDP dan TCP adalah untuk memperluas layanan pengiriman IP antara dua sistem ujung ke layanan pengiriman antara dua proses yang berjalan pada sistem akhir. Memperluas pengiriman host-ke-host ke pengiriman proses-ke-proses disebut transport-layer multiplexing dan demultiplexing. UDP dan TCP juga menyediakan pemeriksaan integritas dengan memasukkan bidang deteksi kesalahan di header segmennya. Dua layanan lapisan transport minimal ini - pengiriman data proses-ke-proses dan pengecekan kesalahan - adalah dua layanan yang disediakan oleh UDP! Khususnya, seperti IP, UDP adalah layanan

yang tidak dapat diandalkan itu tidak menjamin bahwa data yang dikirim oleh satu proses akan tiba utuh ke proses tujuan.

Di sisi lain, TCP menawarkan beberapa layanan tambahan untuk aplikasi. Pertama dan terpenting, ini menyediakan transfer data yang andal. Menggunakan kontrol aliran, nomor urut, ucapan terima kasih, dan penghitung waktu (teknik yang akan kita eksplorasi secara rinci dalam bab ini), TCP memastikan bahwa data dikirim dari proses pengiriman ke proses penerimaan, dengan benar dan teratur. TCP dengan demikian mengubah layanan IP yang tidak dapat diandalkan antara sistem akhir menjadi layanan transportasi data yang andal antar proses. TCP juga menyediakan kontrol kemacetan. Secara longgar, kontrol kemacetan TCP mencegah koneksi TCP mana pun dari swamping link dan router antara host komunikasi dengan jumlah lalu lintas yang berlebihan. TCP berusaha untuk memberikan setiap koneksi yang melintasi tautan yang padat bagian yang sama dari lebar pita tautan. Ini dilakukan dengan mengatur kecepatan di mana sisi pengirim koneksi TCP dapat mengirimkan lalu lintas ke jaringan. Lalu lintas UDP, di lain pihak, tidak diatur. Aplikasi yang menggunakan transportasi UDP dapat mengirim dengan cara apa pun yang diinginkan, selama diinginkan.

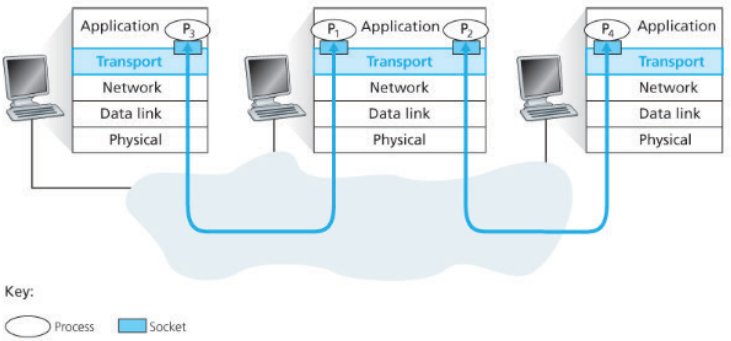
B. Multiplexing dan Demultiplexing

Pada bagian ini, kita membahas multiplexing transport-layer dan demultiplexing, yaitu, memperluas layanan pengiriman host-to-host yang disediakan oleh lapisan jaringan ke layanan pengiriman proses-ke-proses untuk aplikasi yang berjalan di host.

Demultiplexing adalah pekerjaan mengirimkan data dalam segmen lapisan transport ke soket yang benar. Pekerjaan mengumpulkan potongan data pada host sumber dari soket yang berbeda, merangkum setiap potongan data dengan informasi header (yang nantinya akan digunakan dalam demultiplexing) untuk membuat segmen, dan meneruskan segmen ke lapisan jaringan disebut multiplexing.

Transport-layer multiplexing mensyaratkan (1) bahwa soket memiliki pengidentifikasi unik, dan (2) bahwa setiap segmen memiliki bidang khusus yang menunjukkan soket yang akan dikirimi segmen tersebut.

Bidang khusus ini adalah bidang nomor port sumber dan bidang nomor port tujuan. Setiap nomor port adalah nomor 16-bit, mulai dari 0 hingga 65535. Nomor port mulai dari 0 hingga 1023 disebut dengan baik nomor port yang dikenal dan dibatasi, yang berarti mereka dicadangkan untuk digunakan oleh protokol aplikasi yang terkenal seperti HTTP (yang menggunakan nomor port 80) dan FTP (yang menggunakan nomor port 21). Daftar nomor port terkenal diberikan dalam RFC 1700 dan diperbarui di http://www.iana.org [RFC 3232]. Seharusnya sekarang menjadi jelas bagaimana lapisan transport dapat mengimplementasikan layanan demultiplexing: Setiap soket di host dapat diberi nomor port, dan ketika sebuah segmen tiba di host, layer transport memeriksa nomor port tujuan di segmen dan mengarahkan segmen ke soket yang sesuai. Data segmen kemudian melewati soket ke proses terlampir. Seperti yang akan kita lihat, ini pada dasarnya bagaimana UDP melakukannya. Namun, kita juga akan melihat bahwa multiplexing / demultiplexing dalam TCP masih lebih halus.



Gambar 3.2 Multiplexing dan Demultiplexing

- *Multiplexing dan Demultiplexing tanpa Koneksi*

  Python yang berjalan di host dapat membuat soket UDP dengan garis clientSocket = socket (AF_INET, SOCK_DGRAM). Ketika soket UDP dibuat dengan cara ini, lapisan transport secara otomatis memberikan nomor port ke soket. Secara khusus, layer transport memberikan nomor port dalam kisaran 1024 hingga 65535 yang saat ini tidak digunakan oleh port UDP lain di host. Atau, kita dapat menambahkan baris ke program Python setelah kita membuat soket untuk mengaitkan nomor port tertentu (katakanlah, 19157) ke soket UDP ini melalui metode bind socket ():

clientSocket.bind (('', 19157))

Jika pengembang aplikasi yang menulis kode menerapkan sisi server dari "protokol terkenal," maka pengembang harus menetapkan nomor port terkenal yang sesuai. Biasanya, sisi klien aplikasi memungkinkan lapisan transport secara otomatis (dan transparan) menetapkan nomor port, sedangkan sisi server aplikasi menetapkan nomor port tertentu.

Soket UDP sepenuhnya diidentifikasi oleh dua-tupel yang terdiri dari alamat IP tujuan dan nomor port tujuan. Sebagai akibatnya, jika dua segmen UDP memiliki alamat IP sumber dan / atau nomor port sumber yang berbeda, tetapi memiliki alamat IP tujuan dan nomor port tujuan yang sama, maka kedua segmen tersebut akan diarahkan ke proses tujuan yang sama melalui soket tujuan yang sama.

- *Multiplexing dan Demultiplexing Berorientasi Koneksi*

  Untuk memahami TCP demultiplexing, kita harus mencermati soket TCP dan koneksi TCP. Satu perbedaan halus antara soket TCP dan soket UDP adalah soket TCP diidentifikasi oleh empat tupel: (alamat IP sumber, nomor port sumber, alamat IP tujuan, nomor port tujuan). Jadi, ketika segmen TCP tiba dari jaringan ke host, host menggunakan keempat nilai untuk mengarahkan (demultiplex) segmen ke soket yang sesuai. Khususnya, dan berbeda dengan UDP, dua segmen TCP yang datang dengan alamat IP sumber yang berbeda atau nomor port sumber akan (dengan pengecualian segmen TCP yang membawa permintaan pendirian-koneksi yang asli) diarahkan ke dua soket yang berbeda.

  Aplikasi server TCP memiliki "socket sambutan", yang menunggu permintaan koneksi-pendirian dari klien TCP pada nomor port 12000. Klien TCP membuat socket dan mengirimkan segmen permintaan pembentukan koneksi dengan baris-baris:

  clientSocket = socket (AF_INET, SOCK_STREAM) clientSocket. connect ((serverName, 12000))

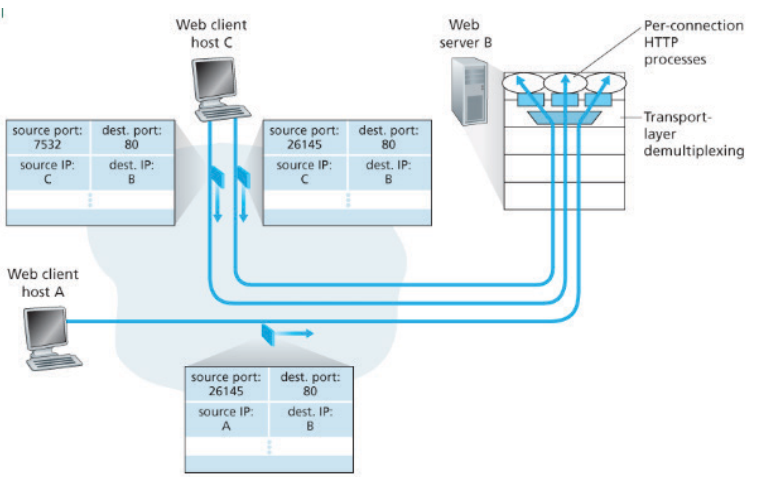
  Permintaan pembentukan koneksi tidak lebih dari segmen TCP dengan nomor port tujuan 12000 dan bit pembentukan koneksi khusus yang diatur dalam header TCP (dibahas dalam Bagian 3.5). Segmen ini juga mencakup nomor port sumber yang dipilih oleh klien. Ketika sistem operasi host dari komputer yang menjalankan proses server

menerima yang masuk segmen permintaan-sambungan dengan port tujuan 12000, ini menempatkan proses server yang sedang menunggu untuk menerima koneksi pada nomor port 12000. Proses server kemudian membuat soket baru:

connectionSocket, addr = serverSocket.accept ()

Lapisan transport di server mencatat empat nilai berikut dalam segmen permintaan koneksi: (1) nomor port sumber di segmen, (2) alamat IP host sumber, (3) nomor port tujuan di segmen, dan (4) alamat IP-nya sendiri. Soket koneksi yang baru dibuat diidentifikasi oleh empat nilai ini; semua segmen yang tiba selanjutnya yang port sumbernya, alamat IP sumber, port tujuan, dan alamat IP tujuan yang cocok dengan keempat nilai ini akan didemultipleksikan ke soket ini. Dengan koneksi TCP sekarang di tempat, klien dan server sekarang dapat saling mengirim data.

Host server dapat mendukung banyak soket koneksi TCP simultan, dengan masing-masing soket terpasang pada suatu proses, dan dengan masing-masing soket diidentifikasi oleh empat tupelnya sendiri. Ketika segmen TCP tiba di host, keempat bidang (alamat IP sumber, port sumber, alamat IP tujuan, port tujuan) digunakan untuk mengarahkan (demultiplex) segmen ke soket yang sesuai.



Gambar 3.3 Multiplexing dan Demultiplexing Berorientasi Koneksi

- *Server Web dan TCP*

  Pertimbangkan host yang menjalankan server Web, seperti server Web Apache, pada port 80. Ketika klien (misalnya, browser) mengirim segmen ke server, semua segmen akan memiliki port tujuan 80. Secara khusus, keduanya adalah koneksi awal yang dibuat. segmen dan segmen yang membawa pesan permintaan HTTP akan memiliki port tujuan 80. Server membedakan segmen dari klien yang berbeda menggunakan alamat IP sumber dan port sumber angka. Jika klien dan server menggunakan HTTP persisten, maka selama durasi koneksi persisten klien dan server bertukar pesan HTTP melalui soket server yang sama.

C. Transport Tanpa Koneksi: UDP

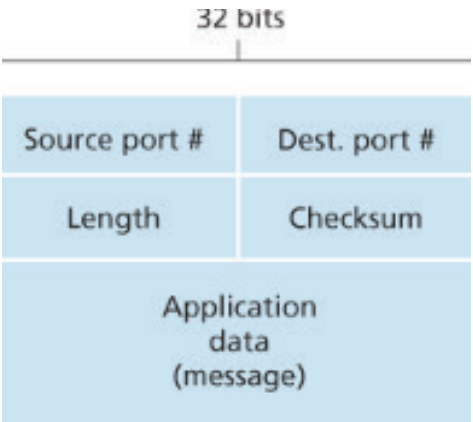
UDP, didefinisikan dalam [RFC 768], hanya melakukan sedikit yang dapat dilakukan oleh protokol transport. Selain dari fungsi multiplexing / demultiplexing dan beberapa pengecekan error ringan, ia tidak menambahkan apa-apa ke IP. Bahkan, jika pengembang aplikasi memilih UDP daripada TCP, maka aplikasi tersebut hampir langsung berbicara dengan IP. UDP mengambil pesan dari proses aplikasi, melampirkan bidang nomor port sumber dan tujuan untuk layanan multiplexing / demultiplexing, menambahkan dua bidang kecil lainnya, dan meneruskan segmen yang dihasilkan ke lapisan jaringan. Lapisan jaringan merangkum segmen lapisan transport ke dalam datagram IP dan kemudian membuat upaya-upaya terbaik untuk mengirimkan segmen ke host penerima. Jika segmen tiba di host penerima, UDP menggunakan nomor port tujuan untuk mengirimkan data segmen ke proses aplikasi yang benar. Perhatikan bahwa dengan UDP tidak ada jabat tangan antara mengirim dan menerima entitas lapisan transportasi sebelum mengirim segmen. Untuk alasan ini, UDP dikatakan tanpa koneksi.

Beberapa alasan pengembang aplikasi memilih untuk membangun aplikasi melalui UDP, karena beberapa aplikasi cocok untuk UDP. Berikut penjelasannya:

- Kontrol tingkat aplikasi yang lebih baik atas data apa dan kapan data dikirim.

- Tidak ada pembentukan koneksi. UDP tidak memperkenalkan penundaan apapun untuk membuat sambungan.
- Tidak ada status koneksi. UDP aktif di sisi lain, tidak mempertahankan ststus koneksi dan tidak melacak parameter ini. Oleh karena itu, server yang dikhususkan untuk aplikasi tertentu biasanya dapat mendukung lebih banyak aplikasi aktif klien ketika aplikasi berjalan melalui UDP daripada TCP.
- *Overhead header* paket kecil. Segmen UDP hanya memiliki 8 byte *overhead***.**

1. Struktur Segmen UDP



Gambar 3.4 Struktur Segmen UDP

Data aplikasi menempati bidang data segmen UDP. Misalnya, untuk DNS, bidang data berisi pesan kueri atau pesan respons. Untuk aplikasi audio streaming, sampel audio mengisi bidang data. Header UDP hanya memiliki empat bidang, masing-masing terdiri dari dua byte. Seperti dibahas di bagian sebelumnya, nomor port memungkinkan host tujuan untuk meneruskan data aplikasi ke proses yang benar berjalan pada sistem tujuan akhir (yaitu, untuk melakukan fungsi demultiplexing).

2. Checksum UDP

Checksum UDP menyediakan deteksi kesalahan. Yaitu, checksum digunakan untuk menentukan apakah bit dalam segmen UDP telah diubah (misalnya, oleh noise di tautan atau saat disimpan di router) saat dipindahkan dari sumber ke tujuan. UDP di sisi pengirim

melakukan pelengkap 1s dari jumlah semua kata 16-bit di segmen tersebut, dengan setiap luapan yang ditemui selama jumlah yang dibungkus. Hasil ini diletakkan di bidang checksum segmen UDP. Di sini kita memberikan contoh sederhana dari perhitungan checksum. Anda dapat menemukan detail tentang implementasi perhitungan yang efisien di RFC 1071 dan kinerja data nyata di [Stone 1998; Stone 2000]. Sebagai contoh, misalkan kita memiliki tiga kata 16-bit berikut:

0110011001100000

0101010101010101

1000111100001100

Jumlah dari dua kata pertama dari 16-bit ini adalah

0110011001100000

0101010101010101 1011101110110101

Menambahkan kata ketiga ke jumlah di atas memberi

1011101110110101

1000111100001100 0100101011000010

Perhatikan bahwa penambahan terakhir ini telah meluap, yang dibungkus. Komplemen 1s diperoleh dengan mengubah semua 0s ke 1s dan mengubah semua 1s menjadi 0s. Jadi komplemen 1s dari jumlah 0100101011000010 adalah 1011010100111101, yang menjadi checksum. Di penerima, keempatnya 16 ditambahkan sedikit kata, termasuk checksum. Jika tidak ada kesalahan yang dimasukkan ke dalam paket, maka jelas jumlah pada penerima adalah 1111111111111111. Jika salah satu bit adalah 0, maka kita tahu bahwa kesalahan telah dimasukkan ke dalam paket.

Ini mengakhiri diskusi kita tentang UDP. Kita akan segera melihat bahwa TCP menawarkan transfer data yang dapat diandalkan ke aplikasinya serta layanan lain yang tidak ditawarkan UDP. Secara alami, TCP juga lebih kompleks daripada UDP. Namun, sebelum membahas TCP, akan berguna untuk mundur dan terlebih dahulu membahas prinsip-prinsip dasar transfer data yang andal.

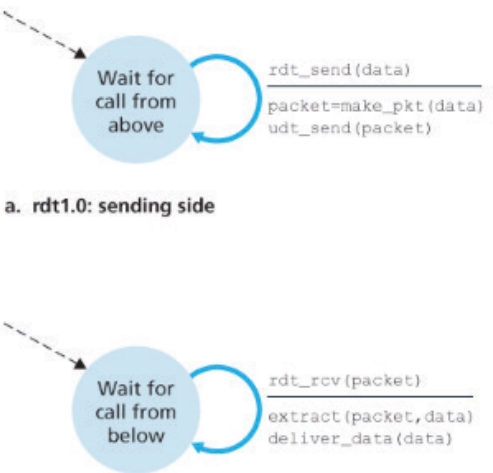
D. Prinsip-Prinsip Transfer Data Yang Handal

Transfer data yang handal terjadi tidak hanya pada lapisan transport, tetapi juga pada layer link dan layer aplikasi. Lapisan atas adalah lapisan yang dapat dipercaya untuk mentransfer data. Dengan saluran yang andal, tidak ada bit data yang ditransfer yang rusak atau hilang, dan semua dikirim sesuai urutan pengirimannya.

Pada bagian ini, yang dipertimbangkan adalah transfer data searah, yaitu transfer data dari mengirim ke sisi penerima. Meski begitu, sisi pengirim dan penerima protokol tetap perlu mengirimkan paket di kedua arah.

1. Membangun Protokol Transfer Data yang Handal

› Transfer Data yang Handal Melalui Saluran yang Sangat Handal: rdt 1.0



Gambar 3.5 Transfer Data Menggunakan rdt 1.0

Dalam prakteknya, rdt_send (data) akan dihasilkan dari panggilan prosedur rdt_send oleh aplikasi lapisan atas. Di sisi penerima, rdt menerima paket dari saluran yang mendasari melalui acara rdt_rcv (paket). Rdt adalah singkatan dari *protocol transder data*.

Menghapus data dari paket melalui ekstrak tindakan paket, data dan meneruskan ke lapisan atas.

Melalui deliver_data (tindakan data), rdt_rcv akan dihasilkan dari panggilan prosedur.

› Transfer Data Melalui Channel yang dapat diandalkan dengan bit error: rdt 2.0

Pada dasarnya, tiga kapabilitas protokol tambahan diperlukan dalam protokol ARQ untuk menangani adanya kesalahan bit: (1) Deteksi kesalahan, (2) Penerima umpan balik, dan (3) Transmisi ulang.

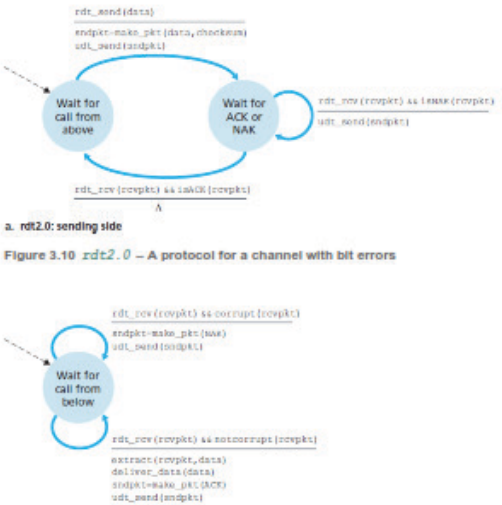
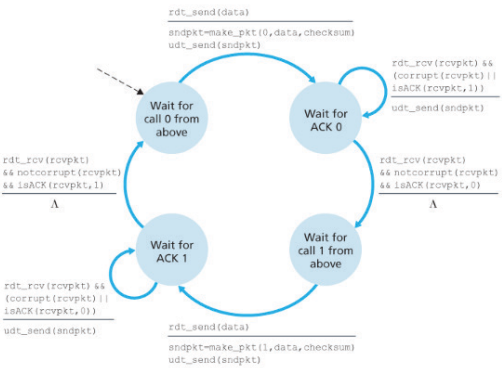


Figure 3.10 *rdt2.0* – A protocol for a channel with bit errors

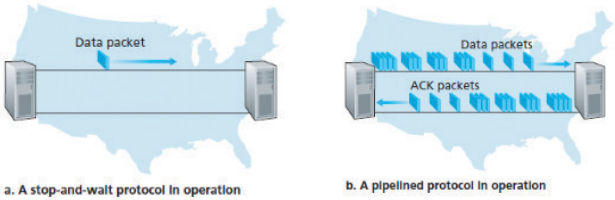
Gambar 3.6 Transfer Data Menggunakan rdt 2.0

› Transfer Data yang Handal Melalui Saluran Kehilangan dengan Kesalahan Bit: rdt 3.0



Gambar 3.7 Transfer Data Menggunakan rdt 3.0

2. Membangun Protokol Data yang Dapat Dilalui



Gambar 3.8 Data Protocol

Kecepatan propaganda round-trip, round-speed antara kedua sistem akhir ini kira-kira 30 milidetik. Misalkan mereka dihubungkan oleh saluran dengan tingkat transmisi, R, 1 Gbps (109 bit per detik). Dengan ukuran paket, L, dari 1.000 byte.

$$D_{trans} = \frac{L}{R} = \frac{8000\ bits/packet}{10\text{^}9\ bits/sec} = 8\ microseconds$$

3. Go-Back-N (GBN)

Dalam protokol Go-Back-N (GBN), pengirim diizinkan untuk mengirimkan beberapa paket (jika tersedia) tanpa menunggu pengakuan, tetapi dibatasi untuk memiliki tidak lebih dari beberapa jumlah maksimum yang diijinkan, N, dari paket yang tidak diakui dalam paket. pipa. Kita menjabarkan protokol GBN secara terperinci di bagian ini. Tetapi sebelum membaca, Anda dianjurkan untuk bermain dengan applet GBN (applet yang luar biasa!) Di situs Web pendamping.

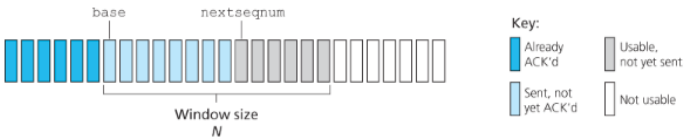


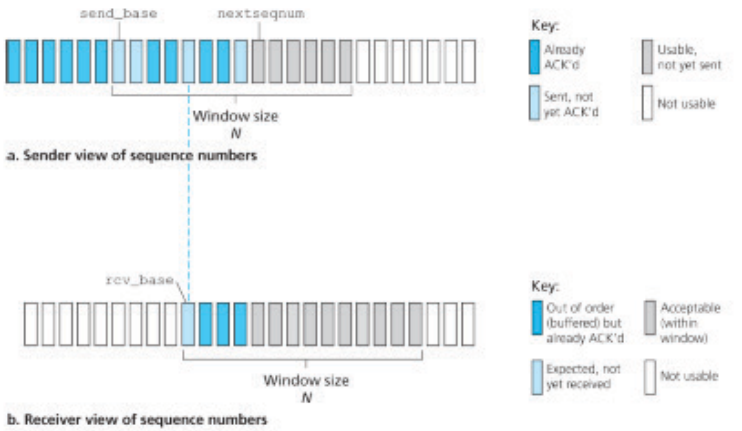
Figure 3.19 Sender's view of sequence numbers in Go-Back-N

Gambar 3.9 GBN Protocol

Kita mencatat di sini bahwa protokol GBN menggabungkan hampir semua teknik yang akan kita temui ketika kita mempelajari komponen transfer data TCP yang dapat diandalkan di Bagian 3.5. Teknik-teknik ini termasuk penggunaan nomor urut, ucapan terima kasih kumulatif, checksum, dan operasi batas waktu / transmisi ulang. n +1

4. Ulangi Selektif (SR)

Protokol GBN memungkinkan pengirim untuk berpotensi "mengisi saluran pipa" dengan paket, sehingga menghindari masalah pemanfaatan saluran yang kita catat dengan protokol stop-and-wait. Kesalahan paket secara keseluruhan dapat menyebabkan GBN mentransmisikan ulang sejumlah besar paket. Namun, ada skenario di mana GBN sendiri mengalami masalah kinerja. Secara khusus, ketika ukuran jendela dan produk bandwidth-delay keduanya besar, banyak paket dapat berada dalam pipa. Dengan demikian, satu paket kesalahan dapat menyebabkan GBN mengirimkan kembali sejumlah besar paket, banyak yang tidak perlu. Ketika probabilitas kesalahan saluran meningkat, pipa dapat diisi dengan transmisi ulang yang tidak perlu ini. Bayangkan, dalam skenario dikte-pesan kita, bahwa jika setiap kali kata dikacaukan, 1.000 kata di sekitarnya (misalnya, ukuran jendela 1.000 kata) harus diulang.



Gambar 3.10 *Selective Repeat*

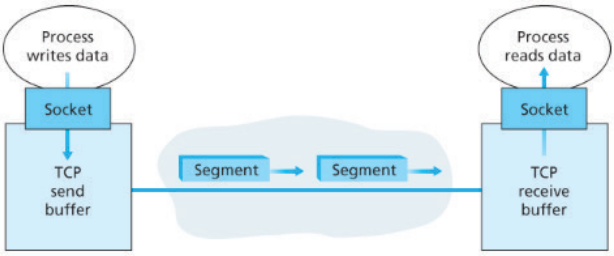
E. Connection-Oriented Transport: TCP

1. Koneksi TCP

TCP dikatakan berorientasi koneksi karena sebelumnya satu proses aplikasi dapat mulai mengirim data di sisi lain, dan harus mengirim beberapa segmen awal satu sama lain untuk menetaplan parameter transfer data berikutnya. Bagian dari pembentukan koneksi TCP,

kedua sisi koneksi akan menginisialisasi banyak variabel status TCP terkait dengan TCP koneksi.

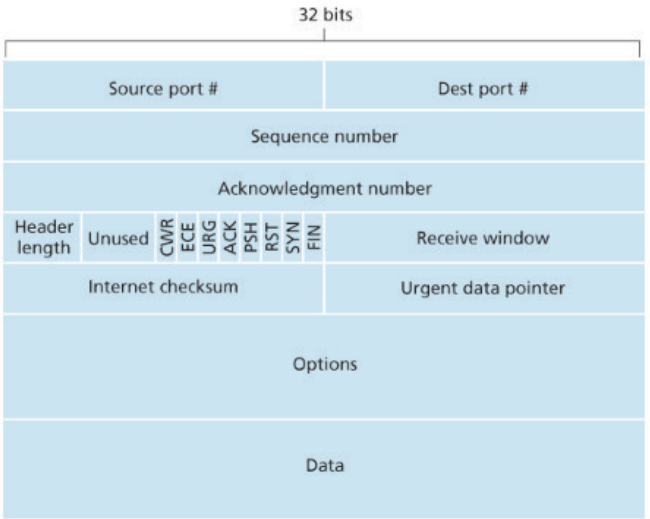
Sambuangan TCP bukanlah rangkaian TDM atau FDM ujung ke ujung seperti pada jaringan saklar-sirkuit. Sebagai gantinya, "koneksi" adalah salah satu yang logis, dengan keadaan umum hanya berada di TCP di dua yang berkomunikasi sistem akhir. Koneksi TCP menyediakan layanan dupleks penuh, TCP koneksi juga selalu point-to-point, yaitu antara satu pengirim dan satu penerima.



Gambar 3.11 TCP Mengirim dan Menerima Buffers

2. Struktur Segmen TCP

TCP segmen terdiri dari bidang tajuk dan bidang data. Bidang data berisi potongan data aplikasi.



Gambar 3.12 Struktur Segmen TCP

3. Estimasi Waktu Pulang Pergi dan Batas Waktu

Batas waktu harus lebih besar daripada waktu pulang-pergi koneksi (RTT) koneksi, yaitu, waktu dari ketika suatu segmen dikirim sampai diakui. Margin harus besar ketika ada banyak fluktuasi dalam nilai-nilai SampleRTT; itu harus kecil ketika ada sedikit fluktuasi.

Memperkirakan Waktu Pulang-Pergi

Sampel RTT, dilambangkan dengan Sampel RTT, untuk segmen adalah jumlah waktu antara saat segmen dikirim dan kapan pengakuan untuk segmen tersebut diterima. Alih-alih mengukur SampelRTT untuk setiap segmen yang ditransmisikan, sebagian besar implementasi TCP hanya mengambil satu pengukuran SampelRTT pada satu waktu. Artinya, pada titik waktu manapun, SampleRTT diestimasi hanya untuk satu tetapi yang ditransmisikan segmen yang saat ini belum diakui, yang mengarah ke nilai baru SampleRTT kira-kira sekali setiap RTT. Juga, TCP tidak pernah menghitung SampleRTT untuk segmen yang telah ditransmisikan ulang; itu saja mengukur SampleRTT untuk segmen yang telah dikirim satu kali.

$$EstimatedRTT=(1-\alpha)\cdot EstimatedRTT+\alpha\cdot SampleRTT$$

EstimatedRTT adalah rata-rata setimbang dari nilai SampleRTT.

$$DevRTT=(1-\beta)\cdot DevRTT+\beta\cdot|SampleRTT-EstimatedRTT|$$

DevRTT adalah EWMA dari perbedaan antara SampleRTT dan EstimatedRTT. Jika nilai SampleRTT memiliki sedikit fluktuasi, maka DevRTT akan kecil; sebaliknya jika jumlahnya banyak fluktuasi, DevRTT akan menjadi besar. Nilai β yang direkomendasikan adalah 0.25. untuk menentukan interval batas waktu transmisi ulang:

$$TimeoutInterval=EstimatedRTT+4\cdot DevRTT$$

Direkomendasikan nilai TimeoutInterval awal 1 detik. Juga saat timeout terjadi, nilai TimeoutInterval digandakan untuk menghindari waktu tunggu prematur yang terjadi untuk segmen selanjutnya yang akan segera diakui. Namun, begitu segmen diterima dan EstimatedRTT diperbarui, TimeoutInterval lagi dihitung menggunakan rumus di atas.

4. Transfer Data Yang Andal

Ingat bahwa layanan lapisan jaringan (layanan IP) Internet tidak dapat diandalkan. IP tidak menjamin pengiriman datagram, tidak menjamin pengiriman datagram secara berurutan, dan tidak menjamin integritas data dalam datagram. Dengan layanan IP, datagram dapat meluap buffer router dan tidak pernah mencapai tujuan mereka, datagram bisa rusak, dan bit dalam datagram bisa rusak (dibalik dari 0 ke 1 dan sebaliknya). Karena segmen layer transport dilakukan di seluruh jaringan oleh datagram IP, segmen layer transport dapat menderita dari masalah ini juga.

Modifikasi yang diusulkan untuk TCP, yang disebut selective acknowledgment [RFC 2018], memungkinkan penerima TCP untuk mengakui segmen out-of-order secara selektif daripada hanya mengakui secara kumulatif segmen terakhir yang diterima dengan benar. Ketika dikombinasikan dengan transmisi ulang selektif — melewatkan transmisi ulang segmen yang telah diakui secara selektif oleh penerima — TCP sangat mirip dengan protokol SR umum kita. Dengan demikian, mekanisme pemulihan kesalahan TCP mungkin paling baik dikategorikan sebagai hibrida dari protokol GBN dan SR.

5. Kontrol Aliran

Ingatlah bahwa host di setiap sisi koneksi TCP menyisihkan buffer penerima untuk koneksi. Ketika koneksi TCP menerima byte yang benar dan berurutan, itu menempatkan data dalam buffer menerima. Proses aplikasi terkait akan membaca data dari buffer ini, tetapi tidak harus pada saat data tiba. Memang, aplikasi penerima mungkin sibuk dengan beberapa tugas lain dan bahkan mungkin tidak mencoba untuk membaca data sampai lama setelah itu tiba. Jika aplikasi relatif lambat dalam membaca data, pengirim dapat dengan mudah meluap buffer menerima koneksi dengan mengirim terlalu banyak data terlalu cepat.

n <NN − 1

n + 1, n + 2,…, N.

n +1

TCP menyediakan layanan kontrol aliran ke aplikasinya untuk menghilangkan kemungkinan pengirim meluap dari buffer penerima.

Kontrol aliran dengan demikian adalah layanan pencocokan kecepatan — mencocokkan kecepatan pengiriman pengirim dengan kecepatan membaca aplikasi penerima. Seperti disebutkan sebelumnya, pengirim TCP juga dapat dibatasi karena kemacetan dalam jaringan IP; bentuk kontrol pengirim ini disebut kontrol kemacetan. Meskipun tindakan yang diambil oleh kontrol aliran dan kemacetan serupa (pelambatan pengirim), mereka jelas diambil karena alasan yang sangat berbeda.

Nmap adalah alat yang ampuh yang dapat "membentuk sambungan" tidak hanya untuk port TCP terbuka, tetapi juga untuk port UDP terbuka, untuk firewall dan konfigurasinya, dan bahkan untuk versi aplikasi dan sistem operasi. Sebagian besar ini dilakukan dengan memanipulasi segmen manajemen koneksi TCP [Skoudis 2006].

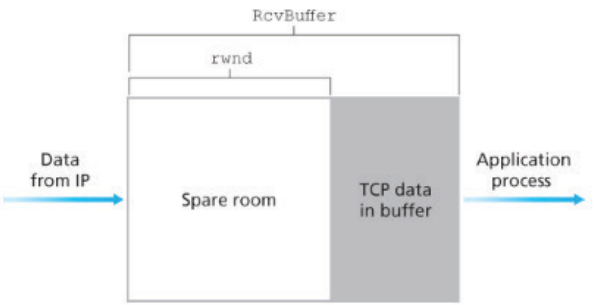


Figure 3.38 The receive window *(rwnd)* and the receive buffer *(RcvBuffer)*

Gambar 3.13 *The Receive(rwnd) dan The Receive Buffer (RcVBuffer)*

F. Prinsip Kontrol Kemacetan

Pada bagian sebelumnya, kita menguji prinsip umum dan mekanisme TCP spesifik yang digunakan untuk menyediakan layanan transfer data yang andal dalam menghadapi hilangnya paket. Kerugian seperti itu biasanya disebabkan oleh melimpahnya buffer router ketika jaringan menjadi padat. Paket transmisi ulang dengan demikian menangani gejala kemacetan jaringan (hilangnya segmen lapisan transport tertentu) tetapi tidak memperlakukan penyebab kemacetan jaringan — terlalu banyak sumber yang mencoba mengirim data dengan laju yang terlalu tinggi. Untuk mengatasi penyebab kemacetan

jaringan, diperlukan mekanisme untuk mencekik pengirim di hadapan kemacetan jaringan.

G. Penyebab dan Biaya Kemacetan

Skenario 1: Dua Pengirim, Perute dengan Buffer Tak Terbatas, tidak ada retransmission, penundaan ketika kemacetan besar, dicapainya throughput maksimal.

Skenario 2: Dua pengirim, sebuah router dengan buffer tak terbatas, pengirim retransmission dari paket yang hilang.

Skenario 3: Empat pengirim, beberapa router dengan buffer tak terbatas, jalur multihop, batas waktu pengiriman ulang.

1. Pendekatan untuk Kontrol Kemacetan

Pada level tertinggi, kita dapat membedakan antara pendekatan kontrol kemacetan dengan apakah layer jaringan memberikan bantuan eksplisit ke layer transport untuk tujuan kontrol kemacetan:

- Kontrol kemacetan ujung ke ujung. Dalam pendekatan ujung ke ujung untuk kontrol kemacetan, lapisan jaringan tidak memberikan dukungan eksplisit ke lapisan pengangkutan untuk tujuan kontrol kemacetan. Bahkan kehadiran kemacetan jaringan harus disimpulkan oleh sistem akhir hanya berdasarkan perilaku jaringan yang diamati (misalnya, packet loss dan delay. Kehilangan segmen TCP dianggap sebagai indikasi kemacetan jaringan, dan TCP mengurangi ukuran jendelanya. Kita juga akan melihat proposal yang lebih baru untuk kontrol kemacetan TCP yang menggunakan peningkatan segmen round-trip delay sebagai indikator peningkatan kemacetan jaringan.
- Kontrol kemacetan yang dibantu jaringan. Dengan kontrol kemacetan yang dibantu jaringan, router memberikan umpan balik eksplisit kepada pengirim dan / atau penerima mengenai keadaan kemacetan jaringan. Umpan balik ini mungkin sesederhana satu bit yang mengindikasikan kemacetan pada suatu tautan - suatu pendekatan yang diambil pada SNA IBM awal [Schwartz 1982], DEC DECnet [Jain 1989; Ramakrishnan 1990] arsitektur, dan arsitektur jaringan ATM [Black 1995].

Umpan balik yang lebih canggih juga dimungkinkan. Misalnya, dalam kontrol kemacetan ATM Available Bite Rate (ABR), router memberi tahu pengirim tentang laju pengiriman host maksimum yang dapat didukung (router) pada tautan keluar.

H. Kontrol Kemacetan TCP

TCP menyediakan layanan transportasi yang andal antara dua proses yang berjalan pada host yang berbeda. Komponen kunci lain dari TCP adalah mekanisme kontrol kemacetannya. TCP harus menggunakan kontrol kemacetan ujung-ke-ujung daripada kontrol kemacetan yang dibantu jaringan, karena lapisan IP tidak memberikan umpan balik eksplisit ke sistem akhir mengenai kemacetan jaringan.

Pendekatan yang diambil oleh TCP adalah membuat setiap pengirim membatasi tingkat pengiriman lalu lintas ke dalam koneksinya sebagai fungsi dari kemacetan jaringan yang dirasakan. Prinsip-prinsip panduan TCP berikut:

- Segmen yang hilang menyiratkan kemacetan, dan karenanya, tingkat pengirim TCP harus dikurangi ketika sebuah segmen hilang.
- Segmen yang menunjukkan bahwa jaringan mengirimkan segmen pengirim ke penerima, dan karenanya, laju pengirim dapat dinaikkan ketika ACK tiba untuk segmen yang sebelumnya tidak diakui.
- Masalah bandwidth

Dengan tinjauan umum tentang kontrol kemacetan TCP ini, kita sekarang dapat mempertimbangkan detail algoritme kontrol-kemacetan TCP yang dirayakan, yang pertama kali dijelaskan dalam [Jacobson 1988] dan distandarisasi dalam [RFC 5681]. Algoritma ini memiliki tiga komponen utama: (1) mulai lambat, (2) menghindari kemacetan, dan (3) pemulihan cepat. Mulai lambat dan penghindaran kemacetan adalah komponen wajib TCP, berbeda dalam cara mereka meningkatkan ukuran cwnd dalam menanggapi ACK yang diterima.

- **Prinsip-Prinsip Dalam Praktek Tcp Splitting: Mengoptimalkan Kinerja Layanan Cloud**

  Untuk layanan cloud seperti pencarian, email, dan jejaring sosial, diinginkan untuk memberikan tingkat responsif yang tinggi, idealnya

memberikan ilusi kepada pengguna bahwa layanan tersebut berjalan di dalam sistem akhir mereka sendiri (termasuk smartphone mereka). Ini bisa menjadi tantangan besar, karena pengguna sering berada jauh dari pusat data yang bertanggung jawab untuk menyajikan konten dinamis yang terkait dengan layanan cloud. Memang, jika sistem akhir jauh dari pusat data, maka RTT akan besar, berpotensi menyebabkan kinerja waktu respons yang buruk karena TCP mulai lambat.

Sebagai studi kasus, pertimbangkan keterlambatan dalam menerima respons untuk permintaan pencarian. Biasanya, server memerlukan tiga jendela TCP selama mulai lambat untuk memberikan respons [Pathak 2010]. Dengan demikian waktu dari ketika sistem akhir memulai koneksi TCP sampai waktu ketika menerima paket terakhir dari respon kira-kira (satu RTT untuk mengatur koneksi TCP ditambah tiga RTT untuk tiga jendela data) ditambah waktu pemrosesan dalam pusat data. Penundaan RTT ini dapat menyebabkan keterlambatan nyata dalam mengembalikan hasil pencarian untuk sebagian kecil kueri. Selain itu, bisa ada kehilangan paket yang signifikan dalam jaringan akses, yang menyebabkan transmisi ulang TCP dan penundaan yang lebih besar.

Salah satu cara untuk mengurangi masalah ini dan meningkatkan kinerja yang dirasakan pengguna adalah (1) menyebarkan server frontend lebih dekat ke pengguna, dan (2) memanfaatkan pemisahan TCP dengan memutus koneksi TCP di server front-end. Dengan pemisahan TCP, klien membuat koneksi TCP ke front-end terdekat, dan front-end mempertahankan koneksi TCP yang persisten ke pusat data dengan jendela kongesti TCP yang sangat besar [Tariq 2008, Pathak 2010, Chen 2011]. Dengan pendekatan ini, waktu respons secara kasar menjadi waktu pemrosesan, di mana RTT adalah waktu bolak-balik antara klien dan server front-end, dan RTT adalah waktu bolak-balik antara server frontend dan pusat data (server back-end). Jika server front-end dekat dengan klien, maka waktu respons ini kira-kira menjadi RTT plus waktu pemrosesan, karena RTT sangat kecil dan RTT kira-kira RTT. Singkatnya, pemisahan TCP dapat mengurangi penundaan jaringan secara kasar dari ke RTT, secara signifikan meningkatkan kinerja yang dirasakan pengguna, terutama bagi pengguna yang jauh dari pusat data terdekat. Pemisahan TCP juga membantu mengurangi keterlambatan pengiriman ulang TCP yang disebabkan oleh kerugian dalam jaringan akses. Google dan Akamai

telah menggunakan server CDN mereka secara ekstensif dalam jaringan akses (ingat diskusi kita di Bagian 2.6) untuk melakukan pemisahan TCP untuk layanan cloud yang mereka dukung [Chen 2011]. Status menghindari kemacetan setelah mengempiskan cwnd. Jika terjadi waktu habis, transisi pemulihan cepat ke kondisi mulai-lambat setelah melakukan tindakan yang sama seperti pada awal yang lambat dan penghindaran kemacetan: Nilai cwnd diatur ke 1 MSS, dan nilai ssthresh diatur ke setengah dari nilai cwnd ketika peristiwa kerugian terjadi.

Pemulihan cepat direkomendasikan, tetapi tidak diperlukan, komponen TCP [RFC 5681]. Sangat menarik bahwa versi awal TCP, yang dikenal sebagai TCP Tahoe, tanpa syarat memangkas jendela kemacetannya menjadi 1 MSS dan memasuki fase mulai-lambat setelah peristiwa kehilangan yang diindikasikan timeout atau triple-duplikat-ACK. Versi terbaru dari TCP, TCP Reno, memasukkan pemulihan cepat.

Dalam gambar ini, ambang awalnya sama dengan 8 MSS. Untuk delapan putaran transmisi pertama, Tahoe dan Reno mengambil tindakan yang identik. Jendela kemacetan naik secara eksponensial cepat selama start lambat dan mencapai ambang batas pada putaran keempat transmisi. Jendela kemacetan kemudian naik secara linear hingga peristiwa tiga rangkap-ACK terjadi, tepat setelah putaran transmisi 8. Perhatikan bahwa jendela kemacetan adalah saat peristiwa kerugian ini terjadi. Nilai ssthresh kemudian diatur ke cwnd Di bawah TCP Reno, jendela congestion diatur ke cwnd dan kemudian tumbuh secara linear. Di bawah TCP Tahoe, jendela kemacetan diatur ke 1 MSS dan tumbuh secara eksponensial hingga mencapai nilai ssthresh, di mana titik itu tumbuh secara linear.

- **Kontrol Kemacetan TCP: Retrospektif**

  Setelah mempelajari rincian mulai lambat, penghindaran kemacetan, dan pemulihan cepat, ada baiknya untuk sekarang mundur dan melihat hutan dari pohon. Mengabaikan

  $12 \cdot \text{MSS}\ 0{,}5 \cdot = 6 \cdot \text{MSS}. = 9 \cdot \text{MSS}$

  Kontrol kemacetan AIMD memunculkan perilaku "gigi gergaji" yang ditunjukkan pada Gambar 3.53, yang juga menggambarkan dengan baik intuisi awal kita tentang "menyelidik" untuk bandwidth

— TCP secara linear meningkatkan ukuran jendela kemacetan (dan karenanya laju transmisi) hingga duplikat rangkap tiga. Peristiwa -ACK terjadi. Ini kemudian mengurangi ukuran jendela kemacetannya dengan faktor dua tetapi sekali lagi mulai meningkatkannya secara linear, memeriksa apakah ada bandwidth tambahan yang tersedia.

1. Keadilan

Pertimbangkan koneksi K TCP, masing-masing dengan jalur ujung ke ujung yang berbeda, tetapi semua melewati jalur penghambat dengan laju transmisi Rbps. (Dengan tautan bottleneck, kita maksudkan bahwa untuk setiap koneksi, semua tautan lain di sepanjang jalur koneksi tidak macet dan memiliki kapasitas transmisi yang melimpah dibandingkan dengan kapasitas transmisi dari tautan bottleneck.) Misalkan setiap koneksi mentransfer file besar dan ada tidak ada lalu lintas UDP yang melewati tautan bottleneck. Mekanisme kontrol kemacetan dikatakan adil jika laju transmisi rata-rata setiap koneksi adalah sekitar R / K;

throughput rata-rata koneksi = 1,22 · MSSRTTL–10

artinya, setiap koneksi mendapat bagian yang sama dari bandwidth tautan.

Apakah algoritma AIMD TCP adil, terutama mengingat bahwa koneksi TCP yang berbeda dapat mulai pada waktu yang berbeda dan dengan demikian mungkin memiliki ukuran jendela yang berbeda pada titik waktu tertentu? [Chiu 1989] memberikan penjelasan yang elegan dan intuitif tentang mengapa kontrol kemacetan TCP menyatu untuk memberikan bagian yang sama dari bandwidth tautan bottleneck di antara koneksi TCP yang bersaing.

› Keadilan dan Koneksi TCP Paralel

Tetapi bahkan jika kita bisa memaksa lalu lintas UDP untuk berperilaku adil, masalah keadilan masih belum sepenuhnya diselesaikan. Ini karena tidak ada yang menghentikan aplikasi berbasis TCP dari menggunakan beberapa koneksi paralel. Misalnya, browser Web sering menggunakan beberapa koneksi TCP paralel untuk mentransfer beberapa objek dalam halaman Web. (Jumlah pasti dari beberapa koneksi dapat dikonfigurasi di sebagian besar browser.) Ketika suatu aplikasi menggunakan

beberapa koneksi paralel, ia mendapatkan fraksi yang lebih besar dari bandwidth dalam tautan yang padat. Sebagai contoh, pertimbangkan tautan tingkat R yang mendukung sembilan aplikasi server klien yang sedang berjalan, dengan masing-masing aplikasi menggunakan satu koneksi TCP. Jika aplikasi baru datang dan juga menggunakan satu koneksi TCP, maka masing-masing aplikasi mendapatkan kira-kira tingkat transmisi R / 10 yang sama. Tetapi jika aplikasi baru ini sebagai gantinya menggunakan 11 koneksi TCP paralel, maka aplikasi baru mendapatkan alokasi yang lebih adil dari lebih dari R / 2. Karena lalu lintas Web begitu meresap di Internet, beberapa koneksi paralel tidak jarang.

2. **Pemberitahuan Kemacetan Eksplisit (ECN): Kontrol Kemacetan yang dibantu jaringan**

Sejak standarisasi awal untuk memulai lambat dan menghindari kemacetan di akhir 1980-an [RFC 1122], TCP telah menerapkan bentuk kontrol kemacetan ujung-ujung yang kita pelajari di Bagian 3.7.1: pengirim TCP tidak menerima indikasi kemacetan eksplisit dari jaringan. lapisan, dan bukannya menyimpulkan kemacetan melalui hilangnya paket yang diamati. Baru-baru ini, ekstensi untuk IP dan TCP [RFC 3168] telah diusulkan, diterapkan, dan digunakan yang memungkinkan jaringan untuk secara eksplisit memberi sinyal kemacetan ke TCP pengirim dan penerima. Bentuk kontrol kongesti berbantuan jaringan ini dikenal sebagai Pemberitahuan Kemacetan Eksplisit. Seperti yang ditunjukkan pada Gambar 3.56, protokol TCP dan IP terlibat.

Protokol transport-layer lain selain TCP juga dapat menggunakan ECN sinyal jaringan-layer. Datagram Congestion Control Protocol (DCCP) [RFC 4340] menyediakan layanan overhead-rendah, kontrol-seperti UDP yang tidak dapat diandalkan seperti kemacetan yang menggunakan ECN. DCTCP (Data Center TCP) [Alizadeh 2010], versi TCP yang dirancang khusus untuk jaringan pusat data, juga memanfaatkan ECN.

I. Ringkasan

Kita memulai bab ini dengan mempelajari layanan yang dapat diberikan oleh protokol transport-layer ke aplikasi jaringan. Pada satu ekstrim, protokol transport-layer bisa sangat sederhana dan menawarkan layanan tanpa embel-embel untuk aplikasi, hanya menyediakan fungsi multiplexing / demultiplexing untuk proses komunikasi. Protokol UDP Internet adalah contoh protokol transport-layer tanpa embel-embel. Pada ekstrem yang lain, protokol transport-layer dapat memberikan berbagai jaminan untuk aplikasi, seperti pengiriman data yang andal, jaminan keterlambatan, dan jaminan bandwidth. Namun demikian, layanan yang dapat disediakan oleh protokol transportasi sering dibatasi oleh model layanan dari protokol lapisan jaringan yang mendasarinya. Jika protokol lapisan jaringan tidak dapat memberikan jaminan keterlambatan atau bandwidth untuk segmen lapisan transportasi, maka protokol lapisan transport tidak dapat memberikan jaminan keterlambatan atau bandwidth untuk pesan yang dikirim di antara proses.

Salah satu dari empat lapisan atas tumpukan protokol dapat menerapkan ucapan terima kasih, pengatur waktu, transmisi ulang, dan nomor urut dan menyediakan transfer data yang andal ke lapisan di atas. Faktanya, selama bertahun-tahun, insinyur dan ilmuwan komputer telah secara independen merancang dan mengimplementasikan protokol tautan, jaringan, transportasi, dan lapisan aplikasi yang menyediakan transfer data yang andal (walaupun banyak dari protokol ini telah menghilang dengan diam-diam). Kita belajar bahwa TCP itu rumit, melibatkan manajemen koneksi, kontrol aliran, dan estimasi waktu pulang pergi, serta transfer data yang andal. Faktanya, TCP sebenarnya lebih kompleks daripada deskripsi kita — kita sengaja tidak membahas berbagai tambalan, perbaikan, dan perbaikan TCP yang diterapkan secara luas di berbagai versi TCP. Namun, semua kompleksitas ini disembunyikan dari aplikasi jaringan. Jika klien pada satu host ingin mengirim data dengan andal ke server di host lain, itu hanya membuka soket TCP ke server dan memompa data ke dalam soket itu. Aplikasi client-server sangat tidak menyadari kompleksitas TCP.

Tanpa kontrol kemacetan, jaringan dapat dengan mudah menjadi macet, dengan sedikit atau tidak ada data yang diangkut dari ujung ke ujung. Dalam Bagian 3.7 kita belajar bahwa TCP mengimplementasikan mekanisme kontrol kemacetan ujung-ke-ujung yang secara aditif

meningkatkan laju transmisi ketika jalur koneksi TCP dinilai bebas kemacetan, dan secara multiplikasi menurunkan laju transmisi ketika kehilangan terjadi. Mekanisme ini juga berusaha untuk memberikan setiap koneksi TCP yang melewati link padat dengan porsi yang sama dari bandwidth tautan. Kita juga memeriksa secara mendalam dampak pembentukan koneksi TCP dan mulai lambat pada latensi. Kita mengamati bahwa dalam banyak skenario penting, pembangunan koneksi dan start lambat berkontribusi signifikan terhadap keterlambatan end-to-end. Kita menekankan sekali lagi bahwa sementara kontrol kemacetan TCP telah berkembang selama bertahun-tahun, itu tetap merupakan bidang penelitian intensif dan kemungkinan akan terus berkembang di tahun-tahun mendatang.

BAB 4
LAPISAN JARINGAN: DATA PLANE

Software Defined Network (SDN) adalah istilah yang merujuk pada konsep/paradigma baru dalam mendesain, mengelola dan mengimplementasikan jaringan, terutama untuk mendukung kebutuhan dan inovasi di bidang ini yang semakin lama semakin kompleks. Konsep dasar SDN adalah dengan melakukan pemisahan eksplisit antara control dan forwarding plane, serta kemudian melakukan abstraksi sistem dan meng-isolasi kompleksitas yang ada pada komponen atau sub-sistem dengan mendefinisikan antar-muka (interface) yang standard.

Dalam konsep SDN, tersedia *open interface* yang memungkinkan sebuah entitas software/aplikasi untuk mengendalikan konektivitas yang disediakan oleh sejumlah sumber-daya jaringan, mengendalikan aliran trafik yang melewatinya serta melakukan inspeksi terhadap atau memodifikasi trafik tersebut.

A. Sekilas tentang Network Layer

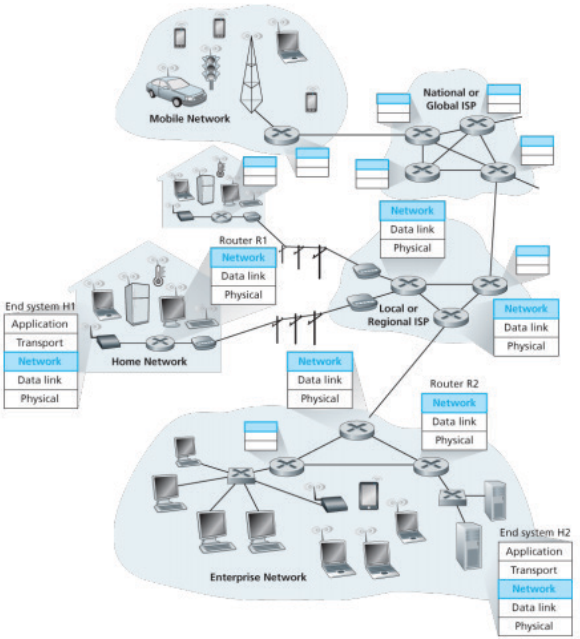
1. Forwarding dan Routing: Data dan Kontrol Pesawat

Yang terakhir merujuk pada proses pemodelan dan abstraksi bidang kontrol jaringan. Menurut Shenker, SDN control plane memerlukan setidaknya 3 jenis abstraksi (SDN v1 & v2) :

a. *Forwarding Abstraction* : bertujuan untuk menjadikan mekanisme *forwarding* yang fleksibel dan tidak bergantung pada jenis perangkat (*vendor neutrality*).

b. *State Distribution Abstraction* : bertujuan untuk medapatkan *global network view* dan menangani semua proses *state dissemination/ collection*. Abstraksi ini dilakukan oleh NOS (*Network Operating System*) yang merupakan sistem terdistribusi, berkomunikasi dengan elemen jaringan untuk membuat *network view*. Aplikasi/ control-program menggunakan *network-view* ini untuk menghasilkan konfigurasi setiap elemen jaringan.

c. *Specification Abstraction* : bertujuan untuk mendapatkan *abstract network view* yang merupakan fungsi dari *global network view*. Abstraksi ini dilakukan oleh *Network Hypervisor* (*Nypervisor)* yang menterjemahkan abstrak ke *global network view*. Dengan *Nypervisor*, aplikasi/control-program dapat berinteraksi dengan jaringan seolah-olah seperti *single-device.*



Gambar 4.1 Lapisan Network

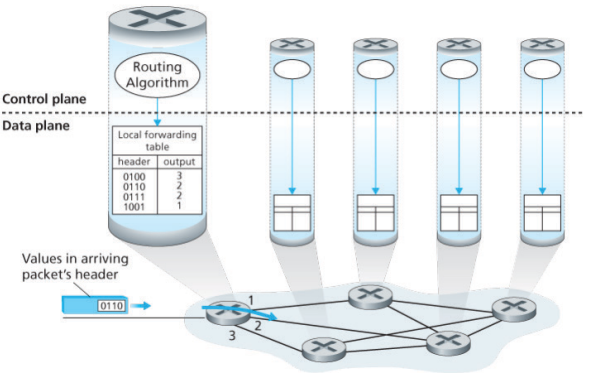
› Lapisan Jaringan

Mengelola pilihan yang berkaitan dengan Host dan Alamat Jaringan, mengelola Sub-Jaringan, dan InterNetworking. Mengambil tanggung jawab untuk Routing Paket dari sumber ke tujuan dalam atau di luar SubNet. Dua subnet yang berbeda

mungkin memiliki skema pengalamatan yang berbeda atau jenis menangani Non-Kompatibel.

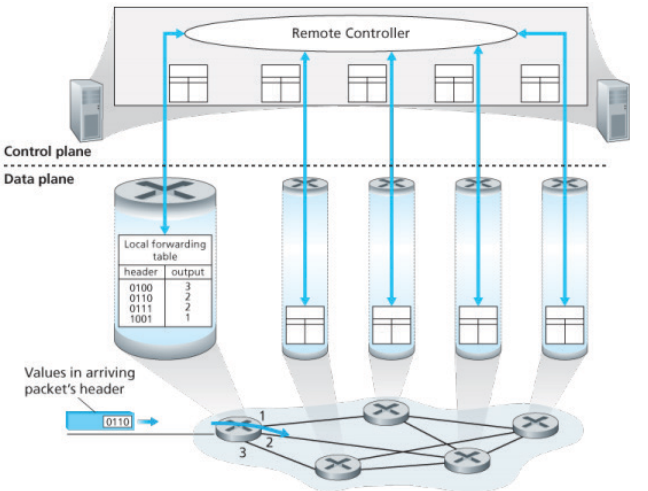
Sama dengan Protokol, dua SubNet yang berbeda beroperasi pada Protokol yang berbeda yang tidak kompatibel satu sama lain. Lapisan jaringan memiliki tanggung jawab untuk rute paket dari sumber ke tujuan, pemetaan skema pengalamatan yang berbeda dan protokol.

Control plane: pendekatan tradisional



Gambar 4.2 Algoritma Perutean Menentukan Nilai dalam Tabel Maju

Control plane: pendekatan SDN



Gambar 4.3 Pengendali Jarak Jauh Menentukan dan Mendistribusikan Nilai dalam Tabel Penerusan

Melalui jaringan, setiap komputer dapat melayani kebutuhan komputer lainnya di dalam jaringan seperti bertukar data mencetak gambar, berbagi resource dan lainnya. Namun, tidak selamanya sebuah komputer menjadi pusat layanan dalam jaringan, tetapi juga membutuhkan bantuan dari komputer lainnya. Oleh karena itu, dilihat dari segi model layanannya, jaringan dibedakan menjadi dua macam, yaitu sebagai berikut.

Mari kita sekarang mempertimbangkan beberapa kemungkinan layanan yang lapisan jaringan dapat menyediakan. Layanan ini dapat Termasuk:

- Dijamin pengiriman.
- Dijamin pengiriman dengan penundaan terbatas.
- Pengiriman paket in-order.
- Bandwidth minimal yang terjamin.
- Keamanan.

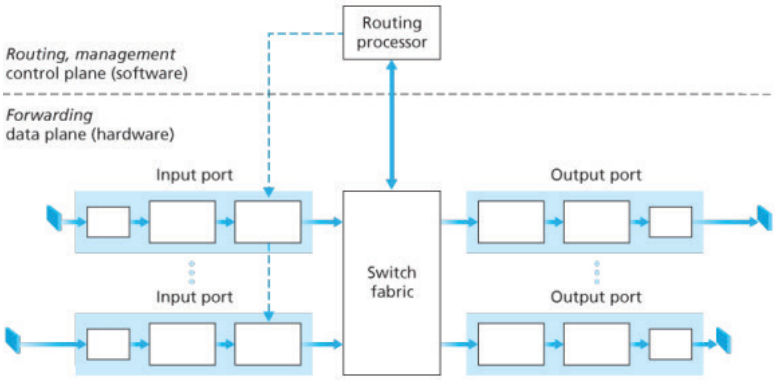
B. Apa yang di Namakan Router?

Router adalah sebuah perangkat yang menghubungkan antara suatu jaringan dengan jaringan lainnya agar bisa berkomunikasi satu sama lain. Mustahil jika di dalam sebuah jaringan jika tidak memiliki router kemudian bisa berhubungan antar jaringan. Ketika kamu melakukan koneksi ke internet baik dirumah dengan menggunakan kabel modem, atau menggunakan hp, pastinya kamu akan melalui sebuah perangkat yang bernama router. Data yang dikirimkan dalam sebuah router memiliki beberapa layer (lapisan) yang biasa disebut dengan OSI Layer (akan dibahas pada artikel berikutnya).

- *Apa yang ada di dalam Router*

  Di dalam router pastinya memiliki informasi mengenai alamat ip untuk semua perangkat yang ada di dalam sebuah jaringan. Misalnya di dalam jaringan komputer kantor, setiap komputer yang terhubung dengan jaringan pasti memiliki yang namanya ip address. Nah tugas dari router ini adalah menyampaikan paket data yang dikirimkan maupun yang diterima oleh komputer tersebut. Sebagai contoh sebuah laptop yang terhubung dengan wifi router, maka ketika laptop

tersebut membuka sebuah alamat website seperti *https://www.google.com* maka tugas dari wifi router adalah menerima data request dari laptop tersebut kemudian menyampaikan kepada isp yang sudah terhubung dengan wifi router yang ada kepada ISP (Internet Service Provider). Kemudian router yang ada di ISP juga diterukan kepada router lainnya hingga sampai pada webserver milik google kemudian memberikan respon kembali sampai dengan laptop tersebut.



Gambar 4.4 Arsitektur Router

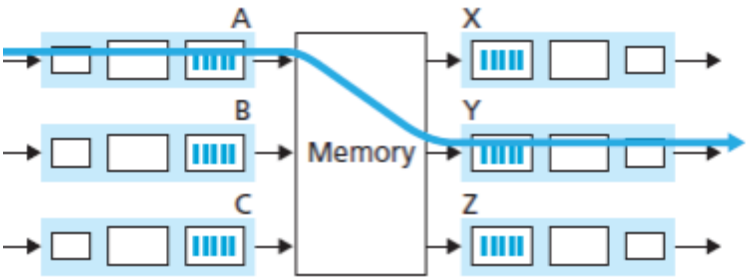
1. Switching

Pengertian paling dasar adalah Switch menciptakan jaringan. Mayoritas jaringan bisnis saat ini menggunakan Network Switch untuk menyambungkan beberapa komputer, telepon, printer, kamera, lampu dan server pada satu lokasi seperti gedung. Network Switch berfungsi sebagai pusat kontrol, memungkinkan semua peralatan untuk saling terhubung dan berfungsi secara efisien. Dengan pembagian data dan informasi serta alokasi sumber daya yang efisien, Network Switch dapat menghemat pengeluaran bisnis anda dan meningkatkan produktifitas para pekerja anda.

Tipe Switch bisa dibagi menjadi 2 tipe; Unmanaged dan Managed. Unmanaged Switch adalah produk Switch yang bisa langsung digunakan tanpa perlu konfigurasi dan instalasi yang rumit. Tentu saja ini berarti juga Unmanaged Switch memiliki keterbatasan fitur dan kapasitas jaringan yang lebih kecil. Di sisi lain adalah Managed Switch. Managed Switch dapat di konfigurasi secara lengkap, menawarkan tingkat keamanan yang lebih tinggi, lebih fleksibel dan kapasitas

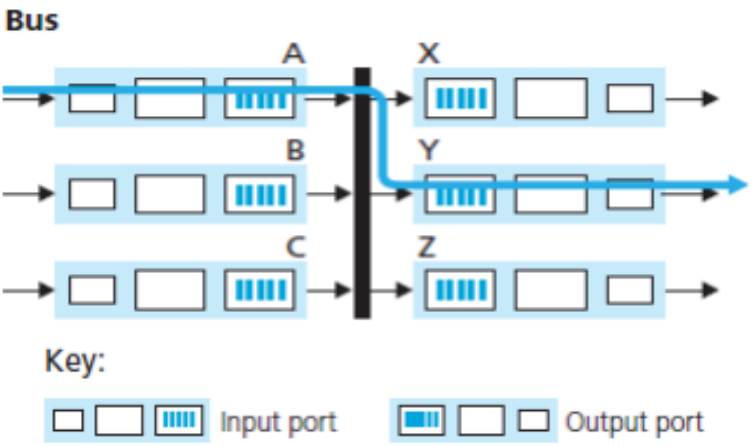
jaringan yang lebih besar. Managed Switch juga dapat dimonitor dan diakses baik secara langsung di lokasi maupun secara remote.

› *Switching Memory*



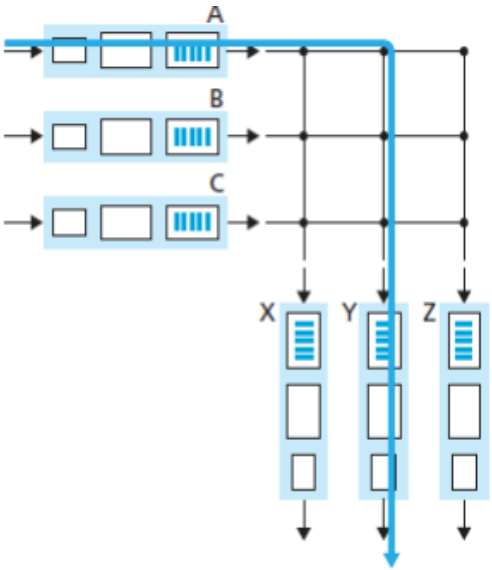
Gambar 4.5 Tiga Teknik *Switching*

› *Switching Bus*



Gambar 4.6 *Switching Bus*

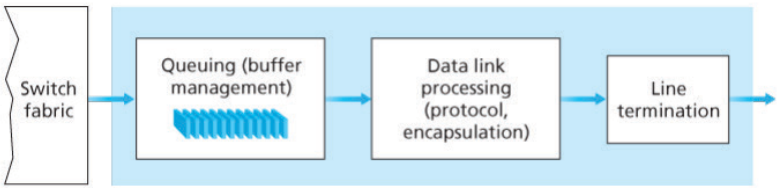
› *Switching Crossbar*



Gambar 4.7 *Switching Crossbar*

2. Di mana antrian terjadi?

Paket antrian dapat terbentuk pada port input dan Port output lihat gambar dibawah ini



Gambar 4.8 Pemrosesan *Port* Keluaran

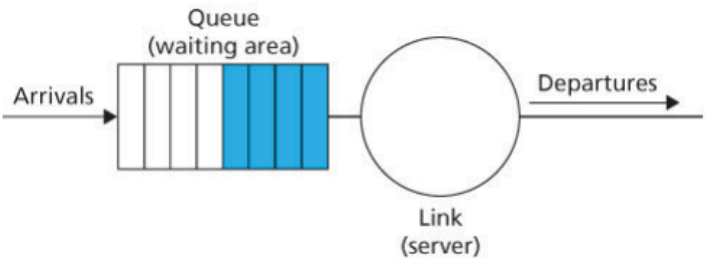
3. Penjadwalan Paket

› First-in-First-Out (FIFO)

Algoritma ini merupakan algoritma penjadwalan yang paling sederhana yang digunakan CPU. Dengan menggunakan algoritma ini setiap proses yang berada pada status ready dimasukkan

kedalam FIFO queue atau antrian dengan prinsip first in first out, sesuai dengan waktu kedatangannya. Proses yang tiba terlebih dahulu yang akan dieksekusi.

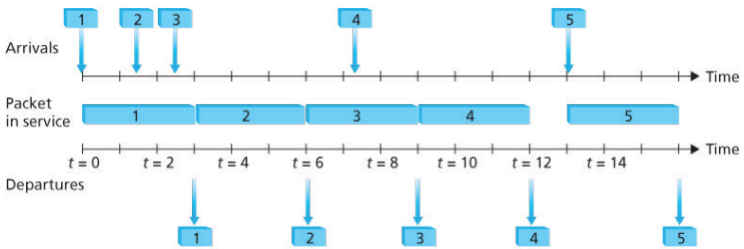
Contoh penjadwalan FIFO



Gambar 4.9 Abstraksi Antrian FIFO

› Antrian Prioritas

Pada layer ini terletak semua aplikasi yang menggunakan TCP/IP ini. Lapisan ini melayani permintaan pemakai untuk mengirim dan menerima data. Data tersebut kemudian disampaikan ke lapisan transport untuk diproses lebihlanjut. Contoh layanan yang diberikan adalah HTTP, FTP, dan SMTP. Contoh antrian prioritas

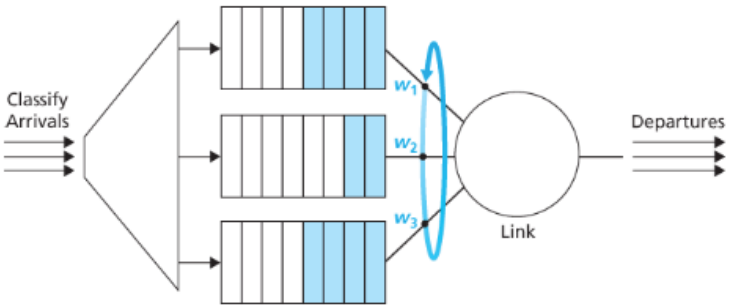


Gambar 4.10 Antrean FIFO sedang beroperasi

› Round Robin dan Weight Fair Queuing (WFQ)

Di bawah disiplin round robin, paket diurutkan ke dalam kelas-kelas seperti dengan antrian prioritas. Namun, daripada ada prioritas layanan yang ketat di antara kelas, penjadwalan round robin layanan alternatif antar kelas. Dalam bentuk paling sederhana dari penjadwalan round robin, paket kelas 1 adalah

ditransmisikan, diikuti oleh paket kelas 2, diikuti oleh paket kelas 1, diikuti oleh paket kelas 2, dan seterusnya. Apa yang disebut disiplin antrean hemat-kerja tidak akan pernah mengizinkan link tetap diam setiap kali ada paket (dari kelas apa pun) yang antri untuk transmisi. Round robin yang menghemat pekerjaan disiplin yang mencari paket kelas tertentu tetapi tidak menemukannya akan segera memeriksa kelas berikutnya di urutan round robin.



Gambar 4.11 Antrian Tertimbang Adil

C. Protokol Internet (IP): IPv4, Pengalamatan, IPv6, dan Lainnya

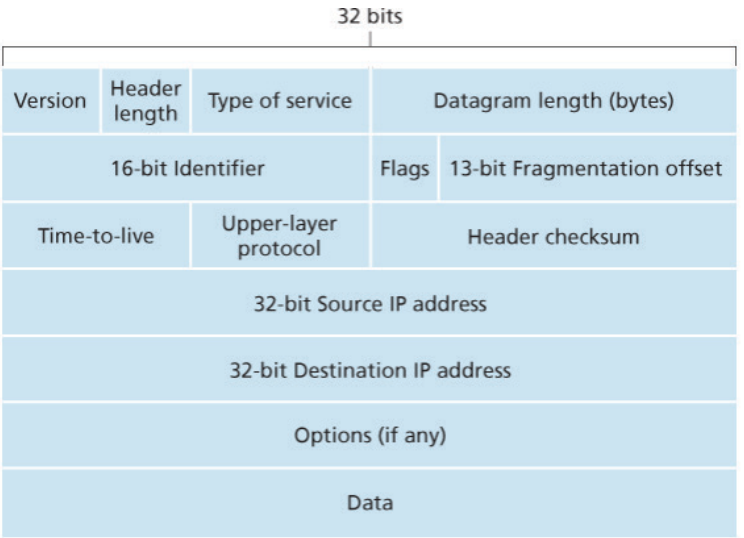
Dalam penulisan IPv6 Anda bisa saja langsung meringkasnya apabila menemui angka nol di depan atau yang biasanya disebut dengan istilah leading zeroes. Apabila terdapat grup angka nol, Anda juga bisa meringkasnya menggunakan teknik double colons. Lalu secara struktur, penulisan alamat IPv6 dibagi menjadi Network prefix dengan interface ID.

Untuk network prefix merupakan alamat yang diberikan oleh RIP (Regional Internet Registry) serta alokasi dari ISP untuk customer. Sedangkan untuk Interface ID adalah pengalamatan kepada sisi host atau perangkat di dalam suatu jaringan. Khusus untuk pengalamatan Interface ID, kita bisa juga menuliskannya secara subnetting.

Paket IPv6 terdiri dari dua bagian yakni paket header dan paket payload. Ukuran dari paket header terdiri dari 40 oktet (320 bit) yang isinya adalah:

- Versi 4 bit
- Traffic class 8 bit
- Label flow 20 bit
- Panjang payload 16 bit
- Header 8 bit
- Batas HOP 8 bit
- Alamat tujuan 128 bit
- Alamat asal 128 bit

Ukuran panjang Payload adalah 166 bit dan bisa juga membawa payload maksimum 65535 oktet.



Gambar 4.12 Antrian Tertimbang Adil

1. Format Datagram IPv4

Paket-paket data dalam protokol IP dikirimkan dalam bentuk datagram. Sebuah datagram IP terdiri atas header IP dan muatan IP (payload), sebagai berikut:

› Header IP: Ukuran header IP bervariasi, yakni berukuran 20 hingga 60 byte, dalam penambahan 4-byte. Header IP menyediakan dukungan untuk memetakan jaringan (routing), identifikasi muatan IP, ukuran header IP dan datagram IP, dukungan fragmentasi, dan juga IP Options.

› Muatan IP: Ukuran muatan IP juga bervariasi, yang berkisar dari 8 byte hingga 65515 byte.

Header dari IP versi 4 terbagi menjadi 14 field yang memiliki fungsi dan informasi yang berbeda, berikut ini adalah sebagai rinciannya :

› **Nomor versi.** 4 bit ini menentukan versi protokol IP dari datagram, dengan ini router dapat menentukan bagaimana menafsirkan sisa dari datagram IP.

› **Panjang header**. 4 bit ini diperlukan untuk menentukan dimana dalam IP datagram playload sebenarnya dimulai.

› **Jenis layanan**. Jenis bit layanan dimasukkan dalam header IPv4 untuk memungkinkan berbeda jenis datagram IP untuk dibedakan satu sama lain.

› **Panjang Datagram**. Ini adalah total panjang datagram IP, diukur dengan format byte.

› **Identifier, flags, fragmentasi mengimbangi**.

› **Time to Live (TTL)**. Disertakan untuk memastikan bahwa datagram tidak bersirkulasi selamanya di jaringan.

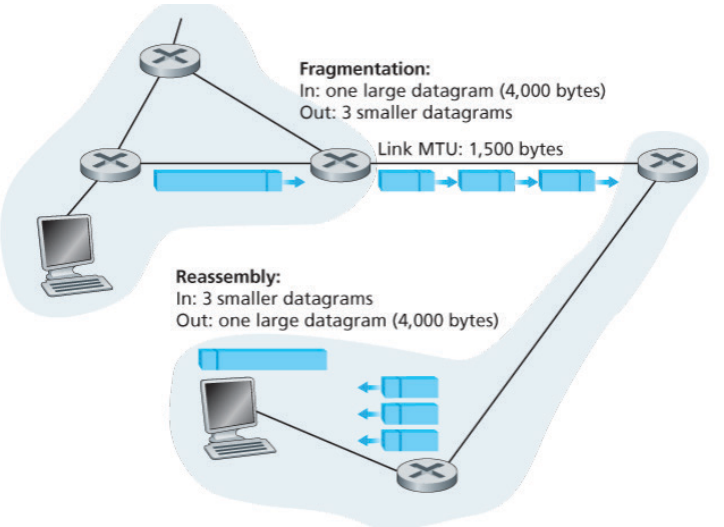
› **Protokol**. Bidang ini biasanya digunakan hanya ketika datagram IP mencapai tujuan akhirnya.

› **Checksum header**. Membantu router dalam mendeteksi kesalahan bit dalam IP yang diterima datagram. Checksum header dihitung dengan memperlakukan setiap 2 byte di header sebagai angka dan menjumlahkan angka-angka ini menggunakan aritmatika komplemen 1s.

› **Alamat IP sumber dan tujuan**. Saat sumber membuat datagram, ia memasukkan alamat IP-nya ke dalam bidang alamat IP sumber dan memasukkan alamat tujuan akhir ke dalam bidang alamat IP tujuan.

› **Pilihan**. Bidang opsi memungkinkan header IP diperpanjang. Opsi tajuk dimaksudkan untuk menjadi jarang digunakan, oleh karena itu keputusan untuk menghemat biaya overhead dengan tidak menyertakan informasi dalam bidang opsi di setiap header datagram.

2. IPv4 datagram fragmentasi



Gambar 4.13 Fragmentasi IP dan Perakitan Ulang

Fragmentasi dan Reassembly telah secara eksklusif dijelaskan dalam RFC 791. Jangan melalui Spesifikasi Protokol Internet RFC. RFC memiliki berbagai bagian yang menjelaskan fragmentasi sampel dan pemasangan kembali. Semua keraguan dan pertanyaan Anda terpenuhi dengan baik di dalamnya.

Jawab 1: Mengenai panjang paket: Paket asli berisi 4000 Bytes. Paket ini adalah paket IP sepenuhnya dan karenanya berisi header IP juga. Jadi panjang payload sebenarnya 4000 - (IP Header Length i. E. 20).

Panjang Payload Aktual = 4000 - 20 = 3980

Sekarang paket tersebut terfragmentasi karena fakta bahwa panjangnya lebih besar dari MTU (1500 Bytes).

Dengan demikian paket 1 berisi 1500 Bytes yang mencakup header IP + Fraksi Payload.

1500 = 20 (header IP) + 1480 (Data Payload)

Demikian pula untuk paket lainnya.

Paket ketiga berisi sisa data yang tersisa (3980 - 1480 -1480) = 1020

Jadi panjang paket adalah 20 (Header IP) + 1020 (payload) = 1040

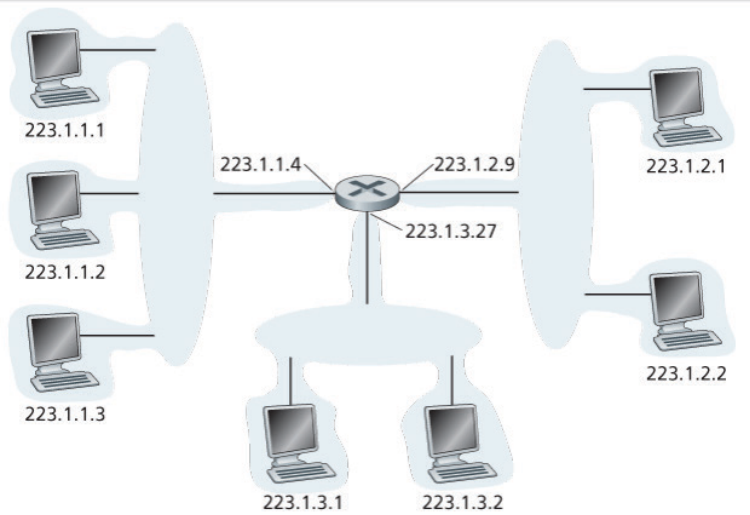
3. Pengalamatan IPv4

Dalam netwoking, pengalamatan IP merupakan hal yang sangat penting karena pengalamatan ini merupakan pengidentifikasian suatu mesin pada jaringan sehingga memiliki identitas yang unik. Untuk tulisan kali ini saya hanya membahas IPv4 (IP vesrsion 4) Pengalamatan IP merupakan pengalamatan untuk jaringan untuk layer 3 pada OSI model.

Alamat IPv4 terdiri dari 32 bit dan ditulis dalam bentuk dotted-decimal. Dotted-decimal adalah penulisan dengan menggunakan "." (titik/dot) sebagai pemisah antara bagian yang satu dengan lainnya, misal 192.168.10.15. Tiap bagian terdiri dari 1 byte (8 bit) dan disebut dengan octet. Pada ipv4 ini, alamat 32 bit ini dipisahkan menjadi 2 bagian yaitu "Alamat Network" (Network portion) dan "Alamat Host" (Host portion). Network portion merupakan identitas dari sekumpulan host dimana hanya yang memeiliki alamat pada host portion yang sama saja host-host dapat saling berkomunikasi. Sedangkan host portion merupakan identitas unik yang dimiliki sebuah mesin yang merupakan identitas dirinya.

Alamat IP adalah nomor 32-bit yang secara unik mengidentifikasi host (komputer atau perangkat lain, seperti printer atau router) pada jaringan TCP/IP. Alamat IP biasanya dinyatakan dalam format desimal bertitik, dengan empat angka dipisahkan oleh titik, seperti 192.168.123.132. Untuk memahami bagaimana Subnet Mask yang digunakan untuk membedakan antara host, Jaringan, dan Subnetwork, periksa alamat IP dalam biner notasi. Sebagai contoh, 192.168.123.132 alamat IP desimal titik (dalam biner notasi) 32 bit nomor 11000000010100011110111000010 0. Angka ini mungkin sulit untuk dimengerti, jadi Bagilah menjadi empat bagian dari delapan Digit biner. Bagian delapan bit ini dikenal sebagai oktet.

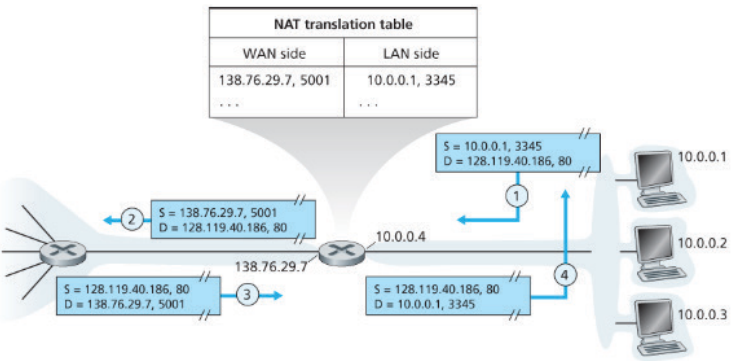
Contoh alamat IP, kemudian, menjadi 11000000.10101000.0111 1011.10000100.



Gambar 4.14 Alamat Antarmuka dan *Subnet*

4. Network Address Translation (NAT)

Network Address Translation (NAT) merupakan sebuah sistem untuk menggabungkan lebih dari satu komputer yang dihubungkan ke dalam jaringan internet hanya dengan menggunakan sebuah alamat IP.



Gambar 4.15 DHCP *Client* dan *Server*

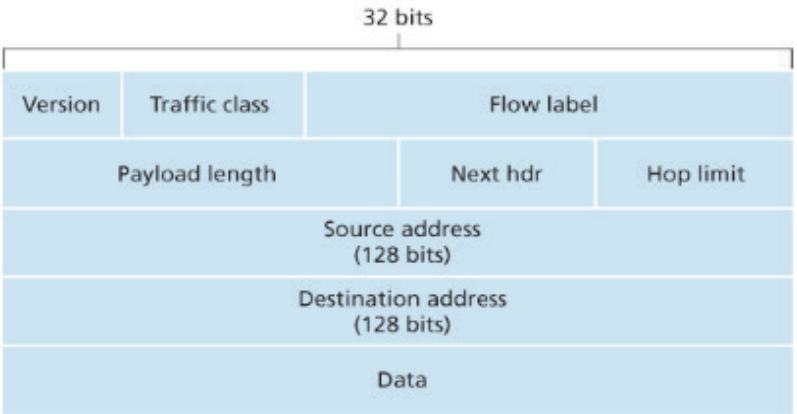
5. IPv6

Pada awal 1990-an, Internet Engineering Task Force memulai upaya untuk mengembangkan penerus Protokol IPv4. Motivasi utama untuk upaya ini adalah kesadaran bahwa ruang alamat IPv4 32-bit

mulai digunakan, dengan subnet dan node IP baru yang terpasang ke Internet (dan sedang dialokasikan alamat IP unik) dengan kecepatan yang menakjubkan. Untuk menjawab kebutuhan ini akan alamat IP yang besar luar angkasa, protokol IP baru, IPv6, dikembangkan.

Format Datagram IPv6

- **Kemampuan pengalamatan yang diperluas**. IPv6 meningkatkan ukuran alamat IP dari 32 menjadi 128 bit.
- **Header 40-byte yang efisien**. Header dengan panjang tetap 40-byte yang dihasilkan memungkinkan pemrosesan IP yang lebih cepat datagram oleh router. Pengodean opsi baru memungkinkan pemrosesan opsi yang lebih fleksibel.
- **Pelabelan aliran**. IPv6 memiliki definisi aliran yang sulit dipahami.



Gambar 4.16 IPv6 *Datagram Format*

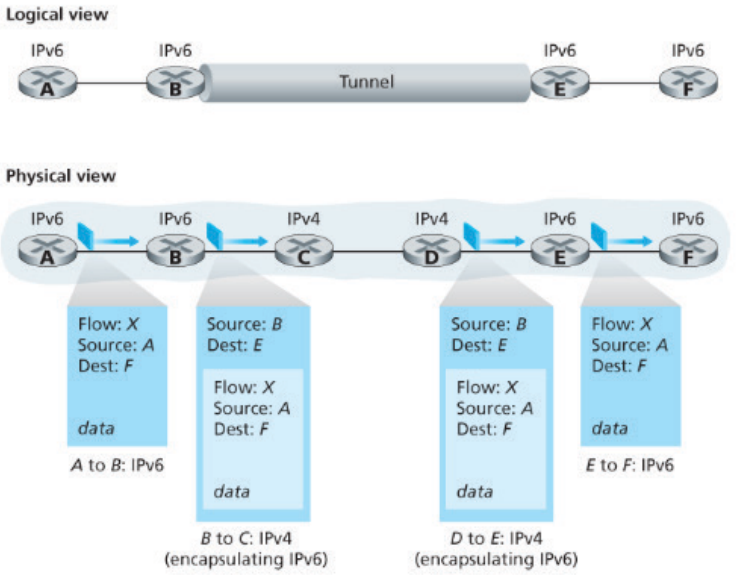
Bidang berikut ditentukan di IPv6:

- **Versi**. Bidang 4-bit ini menunjukkan nomor versi IP. Tak heran, IPv6 mengusung nilai 6 in lapangan ini.
- **Kelas lalu lintas**. Bidang kelas lalu lintas 8-bit, seperti bidang TOS di IPv4, dapat digunakan sebagai prioritas datagram tertentu dalam suatu aliran, atau dapat digunakan untuk memberikan prioritas pada datagram dari tertentu aplikasi (misalnya, voice-over-IP) melalui datagram dari aplikasi lain (misalnya, Email SMTP).

- **Label aliran**. Bidang 20-bit ini digunakan untuk mengidentifikasi aliran datagram.
- **Panjang muatan**. Nilai 16-bit ini diperlakukan sebagai integer tak bertanda tangan yang memberikan jumlah byte dalam file Datagram IPv6 mengikuti header datagram 40-byte dengan panjang tetap.
- **Header berikutnya**. Bidang ini mengidentifikasi protokol yang akan diisi konten (bidang data) datagram ini dikirimkan (misalnya, ke TCP atau UDP). Bidang ini menggunakan nilai yang sama dengan bidang protokol di header IPv4.
- **Batas hop**. Isi bidang ini dikurangi satu per satu oleh setiap router yang meneruskan datagram. Jika jumlah batas hop mencapai nol, datagram-nya adalah dibuang.
- **Data**. Ini adalah bagian payload dari datagram IPv6. Ketika datagram mencapai tujuannya, payload akan dihapus dari datagram IP dan diteruskan ke protokol yang ditentukan di bidang tajuk berikutnya.

Berikut beberapa bidang yang muncul di datagram IPv4 tidak lagi ada di file Datagram IPv6:

- **Fragmentasi / perakitan kembali**. IPv6 tidak mengizinkan fragmentasi dan perakitan ulang di tingkat menengah router; operasi ini hanya dapat dilakukan oleh sumber dan tujuan.
- **Checksum header**. Karena transport-layer (misalnya, TCP dan UDP) dan link-layer (untuk Misalnya, Ethernet) protokol di lapisan Internet melakukan checksumming, fungsi ini cukup berlebihan di lapisan jaringan yang seharusnya dihapus. Pemrosesan cepat paket IP menjadi perhatian utama
- **Pilihan**. Bidang opsi tidak lagi menjadi bagian dari header IP standar. Namun, itu belum hilang jauh. Alih-alih, bidang opsi adalah salah satu dari kemungkinan tajuk berikutnya yang ditunjuk dari dalam IPv6 header.



Gambar 4.17 *Tunneling*

BAB 5
CONTROL PLANE (PESAWAT KONTROL)

Logika Wide-Network yang mengontrol tidak hanya bagaimana datagram diteruskan di antara router sepanjang jalur end-to-end dari host sumber ke host tujuan, tetapi juga bagaimana komponen dan layanan lapisan jaringan dikonfigurasi dan dikelola. kita akan membahas algoritma routing tradisional untuk menghitung jalur biaya paling sedikit dalam sebuah grafik; algoritma ini adalah dasar untuk dua protokol routing internet yang disebarkan secara luas: OSPF dan BGP. OSPF adalah routing Protocol yang beroperasi dalam satu jaringan ISP. BGP adalah protokol routing yang berfungsi untuk interkoneksi semua jaringan di internet; BGP dengan demikian sering disebut sebagai "lem" yang memegang internet bersama-sama.

A. Algoritma Perutean

Tujuan dari algoritma perutean adalah untuk menentukan jalur yang baik (setara, rute), dari pengirim ke penerima, melalui jaringan ruter. Biasanya, jalur "baik" adalah jalur yang memiliki biaya paling sedikit. Klasifikasi perutean algoritma.

- Algoritma routing terpusat menghitung jalur paling murah antara sumber dan tujuan menggunakan mengetahuan global yang lengkap tentang jaringan.
- Algoritma routing desentralisasi, perhitungan lintasan paling murah dilakukan dalam iterative, yang didistribusikan oleh router.

1. Algoritma Perutean Link-State (LS)

Dalam algoritma Link-State, topologi jaringan dan semua biaya tautan diketahui, yaitu tersedia sebagai input untuk algoritma LS. Dalam praktiknya ini dilakukan dengan meminta setiap node menyiarkan paket link-state ke semua node lain dalam jaringan, dengan setiap paket link-state berisi identitas dan biaya tautan yang dilampirkan.

Algoritma perutean LS yang kita sajikan di bawah ini dikenal sebagai algoritma Dijkstra, dinamai menurut penemunya. Algoritma Dijkstra menghitung jalur least-cost (biaya terendah) dari satu node (sumber, yang akan kita sebut sebagai u) ke semua node lain di jaringan. Algoritma Dijkstra bersifat iteratif dan memiliki sifat bahwa setelah iterasi k dari algoritma, jalur least-cost(biaya terendah) diketahui oleh k tujuan node, dan di antara jalur least-cost(biaya terendah) ke semua node tujuan, jalur k ini akan memiliki biaya terkecil.

Algoritma Link-State memiliki beberapa notasi:

- $D(v)$: biaya jalur termurah dari node sumber ke tujuan v pada iterasi algoritma ini
- $p(v)$: node seblumnya (tetangga v) di sepanjang jalur biaya terendah saat ini dari sumber ke v.
- N' : subset dari node; v ada di N' jika jalur berbiaya paling rendag dari sumber ke v diketahui secara pasti.

```
1  Initialization:
2   N' = {u}
3   for all nodes v
4     if v is a neighbor of u
5       then D(v) = c(u, v)
6     else D(v) = ∞
7
8  Loop
9   find w not in N' such that D(w) is a minimum
10  add w to N'
11  update D(v) for each neighbor v of w and not in N':
12        D(v) = min(D(v), D(w)+ c(w, v) )
13   /* new cost to v is either old cost to v or known
14    least path cost to w plus cost from w to v */
15 until N'= N
```

Gambar 5.1 Algoritma Link-State nntuk Node Sumber u

2. Algoritma peruteran Distance-Vector (DV)

Algoritma LS adalah algoritma yang menggunakan informasi global, Sedangkan Algoritma distance-vector (DV) bersifat iteratif, asinkron, dan terdistribusikan. Algoritma perutean DV didistribusikan di mana setiap node menerima beberapa informasi dari satu atau lebih tetangga yang terhubung langsung, melakukan perhitungan, dan kemudian mendistribusikan hasil perhitungannya kembali ke tetangga. Berulang karena proses ini berlanjut sampai tidak ada lagi informasi yang dipertukarkan antar tetangga Algoritma ini tidak sinkron karena tidak memerlukan semua node untuk beroperasi saling berhadapan satu sama lain.

```
Initialization:
  for all destinations y in N:
     Dx(y)= c(x, y)/* if y is not a neighbor then c(x, y)= ∞ */
  for each neighbor w
     Dw(y) = ? for all destinations y in N
  for each neighbor w
     send distance vector  Dx = [Dx(y): y in N] to w

loop
   wait  (until I see a link cost change to some neighbor w or
            until I receive a distance vector from some neighbor w)

   for each y in N:
        Dx(y) = minv{c(x, v) + Dv(y)}

if Dx(y) changed for any destination y
      send distance vector Dx  = [Dx(y): y in N] to all neighbors

forever
```

Gambar 5.2 Algoritma peruteran Distance-Vector

Algoritma Routing Distance-Vector (DV) ini memungkinkan setiap perangkat yang terhubung dalam jaringan dapat membangun dan memelihara tabel routing IP local secara otomatis. Prinsip kerjanya, setiap router pada internetwork menjaga jarak maupun biaya dari router tersebut ke setiap tujuan yang diketahui.

Perbandingan Algoritma Routing LS dan DV

Algoritma DV dan LS mengambil pendekatan pelengkap ke arah komputasi routing. Ingatlah bahwa N adalah himpunan node (router) dan E adalah himpunan tepi (tautan).

› **Kompleksitas pesan**. Kita telah melihat bahwa LS mengharuskan setiap simpul untuk mengetahui biaya setiap tautan dalam jaringan. Ini membutuhkan pesan O (| N | | | |) untuk dikirim. Juga, setiap kali biaya tautan berubah, biaya tautan baru harus dikirim ke semua node. Algoritma DV membutuhkan pertukaran pesan antara tetangga yang terhubung langsung di setiap iterasi. Kita telah melihat bahwa waktu yang dibutuhkan untuk algoritma untuk berkumpul dapat bergantung pada banyak faktor. Ketika biaya tautan berubah, algoritme DV akan menyebarkan hasil dari biaya tautan yang diubah hanya jika biaya tautan baru menghasilkan jalur biaya paling rendah yang diubah untuk salah satu simpul yang terhubung ke tautan itu.

› **Kecepatan konvergensi**. Kita telah melihat bahwa implementasi LS kita adalah algoritma O (| N | 2) yang membutuhkan pesan O (| N | | E |)). Algoritma DV dapat konvergen secara perlahan dan dapat memiliki jalur perutean saat algoritma sedang konvergen. DV juga menderita masalah jumlah hingga tak terbatas.

› **Kekokohan**. Di bawah LS, router dapat menyiarkan biaya yang salah untuk salah satu tautan yang dilampirkan (tetapi tidak ada yang lain). Node juga dapat merusak atau menjatuhkan paket apa pun yang diterimanya sebagai bagian dari siaran LS. Tetapi simpul LS hanya menghitung tabel penerusannya sendiri; node lain melakukan perhitungan serupa untuk diri mereka sendiri. Ini berarti perhitungan rute agak terpisah di bawah LS, memberikan tingkat ketahanan. Di bawah DV, sebuah node dapat mengiklankan jalur berbiaya rendah yang salah ke salah satu atau semua tujuan. Secara lebih umum, kita mencatat bahwa, pada setiap iterasi, perhitungan node dalam DV diteruskan ke tetangganya dan kemudian secara tidak langsung ke tetangga tetangganya pada iterasi berikutnya. Dalam hal ini, perhitungan simpul yang salah dapat disebarkan melalui seluruh jaringan di bawah DV.

B. Perutean Intra-AS di Internet: OSPF

Satu ruter tidak bisa dibedakan dari yang lain dalam arti bahwa semua ruter mengeksekusi algoritma perutean yang sama untuk menghitung jalur perutean melalui seluruh jaringan. Berikut adalah dua alasan penting yang membuat semua yang mengeksekusi algoritma perutean yang sama itu simple:

- **Skala**. Ketika jumlah router menjadi besar, overhead yang terlibat dalam komunikasi, komputasi, dan menyimpan informasi routing menjadi penghalang. Menyimpan informasi perutean untuk kemungkinan tujuan di masing-masing router ini jelas akan membutuhkan memori yang sangat besar. Biaya overhead yang diperlukan untuk menyiarkan konektivitas dan tautan pembaruan biaya di antara semua router akan sangat besar.
- **Otonomi administrative.** Internet adalah jaringan ISP, dengan masing-masing ISP terdiri dari jaringan router sendiri. Suatu ISP umumnya ingin mengoperasikan jaringannya sesuka hati atau untuk menyembunyikan aspek-aspek organisasi internal jaringannya dari luar.

Kedua masalah ini dapat diselesaikan dengan mengatur ruter ke dalam into autonomous systems AS), dengan masing-masing AS terdiri dari sekelompok ruter yang berada di bawah kendali administrasi yang sama. Seringkali ruter di ISP, dan tautan yang menghubungkan mereka, merupakan AS tunggal. Namun, beberapa ISP membagi jaringannya menjadi beberapa AS. Sistem otonom diidentifikasi oleh nomor sistem otonom unik (ASN) globalnya.[RFC 1930]. Nomor AS, seperti alamat IP, ditugaskan oleh pendaftar regional ICANN[ICANN 2016].

Router dalam AS yang sama semua menjalankan algoritma perutean yang sama dan memiliki informasi tentang satu sama lain. Algoritma perutean yang berjalan dalam sistem otonom disebut protokol perutean sistem intra-otonom.

Open Shortest Path First (OSPF)

OSPF adalah protokol link-state yang menggunakan flooding informasi link-state dan algoritma jalur least-cost (biaya terendah) Dijkstra. Dengan OSPF, setiap router membangun peta topologi lengkap (yaitu, grafik) dari

seluruh sistem otonom. Setiap router kemudian secara lokal menjalankan algoritma jalur terpendek Dijkstra untuk menentukan pohon jalur terpendek ke semua subnet, dengan dirinya sebagai simpul akar. Biaya tautan individual dikonfigurasikan oleh administrator Administrator mungkin memilih untuk mengatur semua biaya tautan menjadi 1.

Dengan OSPF, router menyiarkan informasi routing ke semua router lain dalam sistem otonom, tidak hanya ke router tetangganya. Router menyiarkan informasi status tautan setiap kali ada perubahan status tautan (misalnya, perubahan biaya atau perubahan status naik / turun). Itu juga menyiarkan keadaan tautan secara berkala (setidaknya sekali setiap 30 menit), bahkan jika keadaan tautan tidak berubah. RFC 2328 mencatat bahwa "pemutakhiran berkala iklan status tautan ini menambah kekokohan pada algoritma status tautan." Iklan OSPF terkandung dalam pesan OSPF yang dibawa langsung oleh IP, dengan protokol lapisan atas 89 untuk OSPF. Dengan demikian, protokol OSPF itu sendiri harus mengimplementasikan fungsionalitas.

Beberapa kemajuan yang terkandung dalam OSPF meliputi yang berikut:

- **Keamanan.** Pertukaran antara ruter OSPF (misalnya, pembaruan status tautan) dapat disahkan. Dengan otentikasi, hanya router tepercaya yang dapat berpartisipasi dalam protokol OSPF dalam AS, sehingga mencegah penyusup jahat (atau siswa jaringan mengambil pengetahuan yang baru mereka temukan untuk mendapatkan kesenangan) dari menyuntikkan informasi yang salah ke dalam tabel ruter. Secara default, paket OSPF antar router tidak diautentikasi dan bisa dipalsukan.
- **Beberapa jalur dengan biaya yang sama.** Ketika beberapa jalur ke tujuan memiliki biaya yang sama, OSPF memungkinkan banyak jalur untuk digunakan (yaitu, jalur tunggal tidak perlu dipilih untuk membawa semua lalu lintas ketika banyak jalur dengan biaya yang sama).
- **Dukungan terintegrasi untuk perutean unicast dan multicast**. Multicast OSPF (MOSPF) [RFC 1584] memberikan ekstensi sederhana ke OSPF untuk menyediakan routing multicast. MOSPF menggunakan basis data tautan OSPF yang ada dan menambahkan jenis baru tautan-iklan ke mekanisme siaran tautan-OSPF yang ada.

- **Dukungan untuk hierarki dalam AS tunggal**. Sistem otonom OSPF dapat dikonfigurasi secara hierarkis ke area-area. Setiap area menjalankan algoritme perutean status link-OSPF-nya sendiri, dengan setiap router di area menyiarkan link-state-nya ke semua router lain di area itu. Di dalam setiap area, satu atau lebih router perbatasan area bertanggung jawab untuk merutekan paket di luar area. Terakhir, tepat satu area OSPF di AS yang dikonfigurasi untuk menjadi area tulang punggung. Peran utama area tulang punggung adalah untuk mengarahkan lalu lintas antara area lain di AS.

C. Peruteran Antar ISP: BGP

OSPF adalah contoh dari protokol routing intra-AS. Di Internet, semua AS menjalankan protokol perutean antar-AS yang sama, yang disebut Border Gateway Protocol, lebih dikenal sebagai BGP [RFC 4271; Stewart 1999].

BGP bisa dibilang yang paling penting dari semua protokol Internet (satu-satunya pesaing lainnya adalah Protokol IP), karena merupakan protokol yang menempelkan ribuan ISP di Internet bersama. Seperti yang akan kita lihat nanti, BGP adalah protokol terdesentralisasi dan asinkron dalam jalur jarak-vektor. Meskipun BGP adalah protokol yang kompleks dan menantang, untuk memahami Internet secara mendalam, kita perlu mengenal dasar-dasar dan operasinya.

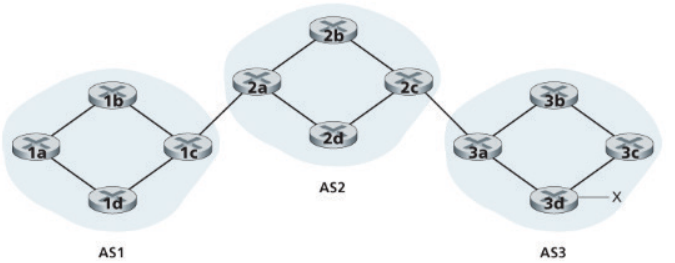
1. Peran BGP

Sebagai protokol routing antar-AS, BGP menyediakan setiap router sarana untuk:

a. **Mendapatkan informasi jangkauan awalan dari AS terdekat.** Secara khusus, BGP memungkinkan setiap subnet untuk mengiklankan keberadaannya ke seluruh Internet.

b. **Menentukan rute "terbaik" ke awalan**. Router dapat belajar tentang dua atau lebih rute berbeda ke awalan tertentu. Untuk menentukan rute terbaik, router akan menjalankan prosedur pemilihan rute BGP secara lokal. Rute terbaik akan ditentukan berdasarkan kebijakan serta informasi jangkauan.

2. Mengiklankan Informasi Rute BGP



Gambar 5.3 Jaringan dengan tiga sistem otonom. AS3 menyertakan subnet dengan awalan x

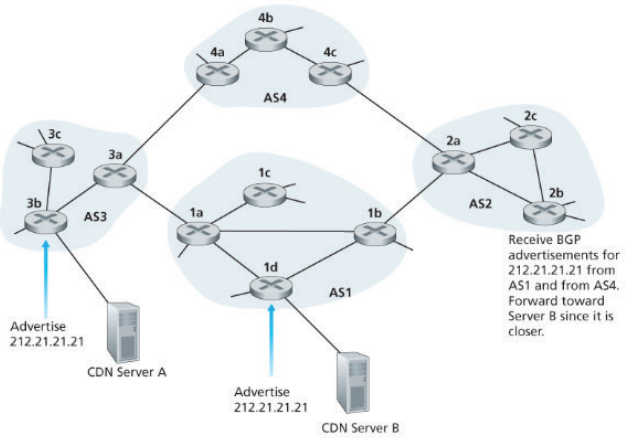
Pada Gambar 5.3, jaringan sederhana ini memiliki tiga otonom sistem: AS1, AS2, dan AS3. AS3 menyertakan subnet dengan awalan x. Untuk setiap AS, setiap router adalah router gateway atau router internal. Router gateway adalah router di AS lain. Router internal hanya terhubung ke host dan router dalam AS-nya sendiri.

Setiap koneksi TCP tersebut, bersama dengan semua pesan BGP yang dikirim melalui koneksi, disebut BGP koneksi. Selanjutnya, koneksi BGP yang menjangkau dua AS disebut BGP eksternal (eBGP) koneksi, dan sesi BGP antara router di AS yang sama disebut BGP internal (iBGP) koneksi. Biasanya satu koneksi eBGP untuk setiap tautan yang secara langsung menghubunglan router gateway di AS yang berbeda.

3. IP-Anycast

Selain menjadi protokol perutean antar-AS internet, BGP sering digunakan untuk mengimplementasikan layanan Ipanycast yang biasanya digunakan dalam DNS. Sistem DNS dapat mereplikasi catatan DNS di seluruh server DNS dunia. Algoritme pemilihan rute BGP memberikan kemudahan dan mekanisme alami untuk melakukannya begitu.

Anycast adalah teknik jaringan tempat awalan IP yang sama diiklankan dari berbagai lokasi. Jaringan kemudian memutuskan lokasi untuk merutekan permintaan pengguna, berdasarkan biaya protokol routing dan mungkin 'kesehatan' dari server iklan.



Gambar 5.4 Menggunakan IP-*anycast* untuk membawa pengguna ke *server* CDN terdekat

D. Pesawat Kontrol SDN

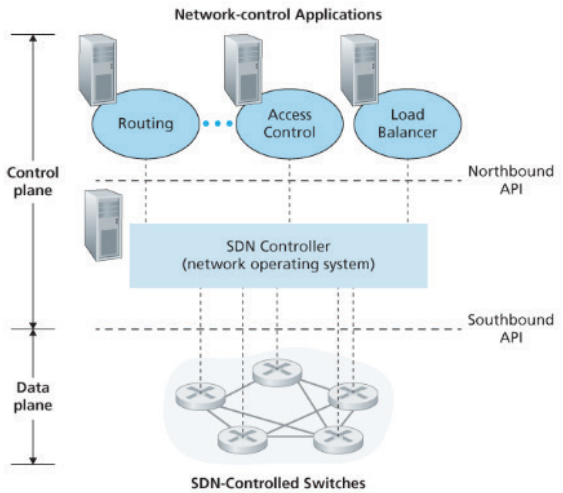
Di bagian ini, kita akan terjun ke bidang kontrol SDN — logika seluruh jaringan yang mengontrol penerusan paket di antara perangkat yang mendukung SDN jaringan, serta konfigurasi dan pengelolaan perangkat ini dan layanannya. Empat karakteristik kunci dari arsitektur SDN dapat diidentifikasi [Kreutz 2015]:

- **Penerusan berbasis aliran.** Penerusan paket dengan sakelar yang dikontrol SDN dapat didasarkan pada sejumlah nilai bidang header dalam header transport-layer, network-layer, atau link-layer.
- **Pemisahan bidang data dan bidang control.** Pesawat data terdiri dari sakelar jaringan— perangkat yang relatif sederhana (tetapi cepat) yang menjalankan aturan "kecocokan plus tindakan" dalam tabel alirannya. Pesawat kendali terdiri dari server dan perangkat lunak yang menentukan dan mengelola tabel aliran sakelar.
- **Jaringan yang dapat diprogram.** Jaringan diprogram melalui aplikasi kontrol jaringan yang berjalan di bidang kontrol. Aplikasi ini mewakili "otak" dari bidang kontrol SDN, menggunakan API yang disediakan oleh pengontrol SDN untuk menentukan dan mengontrol bidang data dalam perangkat jaringan. Aplikasi jaringan lain mungkin melakukan kontrol akses, yaitu, menentukan paket mana yang akan diblokir, Namun

aplikasi lain mungkin meneruskan paket dengan cara yang melakukan penyeimbangan beban server.

- **Fungsi kontrol jaringan: sakelar eksternal ke bidang data.** Mengingat bahwa "S" di SDN adalah untuk "perangkat lunak," mungkin tidak mengherankan bahwa bidang kontrol SDN diimplementasikan dalam perangkat lunak. Tidak seperti router tradisional, perangkat lunak ini dijalankan pada server yang berbeda dan jauh dari switch jaringan. Sebuah pengontrol SDN (atau sistem operasi jaringan) [Gude 2008]) dan satu set aplikasi kontrol jaringan. Pengontrol menjaga informasi keadaan jaringan yang; memberikan informasi ini ke aplikasi kontrol jaringan yang berjalan di bidang kontrol; dan menyediakan cara yang melaluinya aplikasi ini dapat memonitor, memprogram, dan mengontrol perangkat jaringan yang mendasarinya.

Dari diskusi ini, kita dapat melihat bahwa SDN mewakili "unbundling" fungsionalitas jaringan yang signifikan — data switch pesawat, pengontrol SDN, dan aplikasi kontrol jaringan adalah entitas terpisah yang masing-masing dapat disediakan oleh vendor dan organisasi yang berbeda. Ini kontras dengan model pra-SDN di mana switch / router (bersama-sama dengan perangkat lunak bidang kontrol tertanam dan implementasi protokol) adalah monolitik, terintegrasi secara vertikal, dan dijual oleh satu vendor. Pembubaran fungsionalitas jaringan di SDN ini telah disamakan dengan evolusi sebelumnya dari komputer ke komputer pribadi.

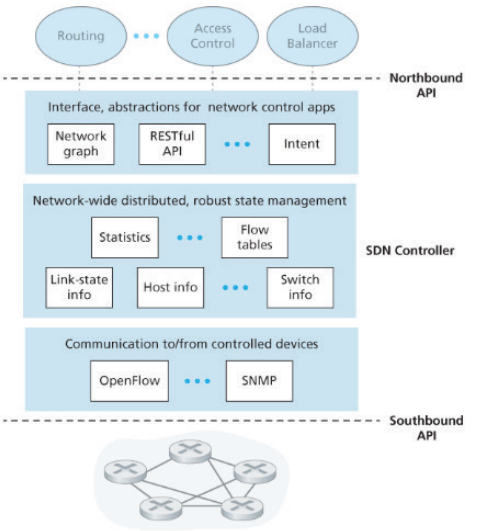


Gambar 5.5 Komponen arsitektur SDN: Sakelar yang dikontrol SDN, pengontrol SDN, dan Aplikasi Kontrol Jaringan

1. **Bidang Kontrol SDN: Pengontrol SDN dan Aplikasi Kontrol Jaringan SDN**

Bidang kontrol SDN terbagi menjadi dua komponen – pengontrol SDN dan aplikasi kontrol jaringan SDN. Sebuah fungsionalitas pengontrol dapat diatur secara luas menjadi tiga lapisan.

- **Lapisan Komunikasi: berkomunikasi antara pengontrol SDN dan jaringan yang dikendalikan perangkat**. Dalam hal ini, protokol diperlukan untuk mentransfer informasi antara pengontrol dan perangkat itu. Selain itu, perangkat harus dapat mengkomunikasikan peristiwa yang diamati secara lokal ke controller.
- **Lapisan manajemen *network state* di seluruh jaringan**. Kendali akhir yang dibuat oleh kendali SDN plane akan membutuhkan file pengontrol yang memiliki informasi terbaru tentang status host jaringan, tautan, sakelar dan perangkat lainnya yang dikontrol SDN.
- **Antarmuka ke lapisan apliksi pengontrol jaringan**. Pengontrol berinteraksi dengan aplikasi kontrol jaringan melalui antarmuka "arah utara". API ini memungkinkan aplikasi kontrol jaringan untuk membaca/menulis status jaringan dan tabel aliran dalam lapisan manajemen status. Berbagai API mungkin disediakan; kita akan melihat bahwa dua pengontrol SDN populer berkomunikasi dengan aplikasi mereka menggunakan antarmuka permintaan-respons.



Gambar 5.6 Komponen pengontrol SDN

2. Protokol OpenFlow

Protokol OpenFlow beroperasi antara pengontrol SDN dan Sakelar yang dikontrol SDN atau perangkat lain yang mengimplementasikan OpenFlow API yang telah kita pelajari sebelumnya. Protokol OpenFlow beroperasi melalui TCP, dengan nomor port default 6653. Di antara pesan penting yang mengalir dari pengontrol ke sakelar terkontrol adalah sebagai berikut:

- **Konfigurasi**. Pesan ini memungkinkan pengontrol untuk menanyakan dan menyetel konfigurasi sakelar parameter.
- **Ubah-Status**. Pesan ini digunakan oleh pengontrol untuk menambah / menghapus atau mengubah entri di sakelar tabel alir, dan untuk mengatur properti port sakelar.
- **Baca-Status**. Pesan ini digunakan oleh pengontrol untuk mengumpulkan statistik dan nilai counter dari port dan tabel aliran sakelar.
- **Kirim-Paket**. Pesan ini digunakan oleh pengontrol untuk mengirim paket tertentu dari yang ditentukan port di sakelar yang dikendalikan. Pesan itu sendiri berisi paket yang akan dikirim dalam payloadnya.

Di antara pesan yang mengalir dari sakelar yang dikontrol SDN ke pengontrol adalah sebagai berikut:

- **Arus-Dihapus**. Pesan ini memberi tahu pengontrol bahwa entri tabel aliran telah dihapus, untuk misalnya dengan batas waktu atau sebagai hasil dari pesan status ubah yang diterima.
- **Status port**. Pesan ini digunakan oleh sakelar untuk memberi tahu pengontrol tentang perubahan status port.
- **Paket masuk**. Paket yang tiba di port switch dan tidak cocok dengan aliran apa pun entri tabel dikirim ke pengontrol untuk pemrosesan tambahan. Paket yang cocok juga dapat dikirim ke pengontrol, sebagai tindakan yang akan dilakukan pada pertandingan. Pesan paket-masuk digunakan untuk mengirim pesan tersebut paket ke pengontrol.

E. ICMP: Protokol Pesan Kontrol Internet

Internet Control Message Protocol (ICMP), digunakan oleh host dan router untuk mengkomunikasikan informasi lapisan jaringan satu sama lain. Penggunaan ICMP paling umum adalah untuk pelaporan kesalahan. Misalnya, saat menjalankan sesi HTTP, Anda mungkin menemukan pesan kesalahan seperti "Jaringan tujuan tidak dapat dijangkau." Pesan ini berasal dari ICMP. Pada titik tertentu, router IP tidak dapat menemukan jalur ke host yang ditentukan dalam permintaan HTTP Anda. Router itu membuat dan mengirim pesan ICMP ke host Anda yang menunjukkan kesalahan.

ICMP sering dianggap sebagai bagian dari IP, tetapi secara arsitektur terletak tepat di atas IP, karena pesan ICMP dibawa di dalam datagram IP. Artinya, pesan ICMP dibawa sebagai muatan IP, seperti halnya segmen TCP atau UDP dibawa sebagai muatan IP.

Pesan ICMP memiliki jenis dan bidang kode, dan berisi header dan 8 byte pertama dari datagram IP yang menyebabkan pesan ICMP dihasilkan di tempat pertama (sehingga pengirim dapat menentukan datagram yang menyebabkan kesalahan).Program ping yang terkenal mengirim pesan ICMP tipe 8 kode 0 ke host yang ditentukan. Host tujuan, melihat permintaan gema, mengirim kembali tipe 0 kode 0 balasan gema ICMP. Pesan ICMP lain yang menarik adalah sumber quench message. Pesan ini jarang digunakan dalam praktik. Tujuan awalnya adalah untuk melakukan kontrol kemacetan — untuk memungkinkan router yang macet mengirim pesan ICMP ke host untuk memaksa host itu untuk mengurangi laju transmisinya.

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

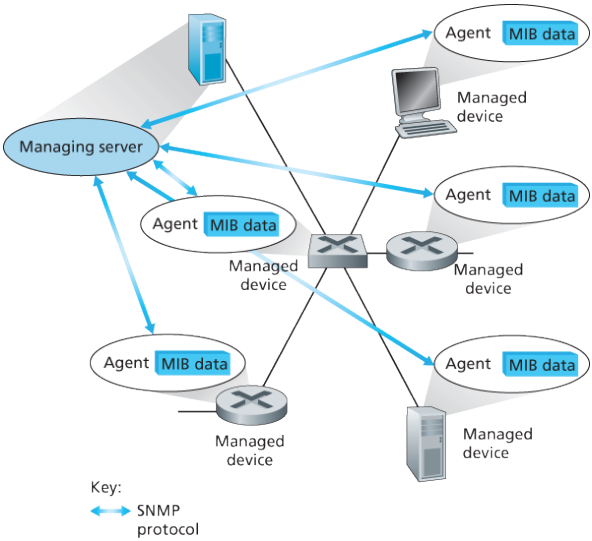
Gambar 5.7 Jenis Pesan ICMP

Versi baru ICMP telah ditetapkan untuk IPv6 di RFC 4443. Selain mengatur ulang tipe ICMP dan definisi kode yang ada, ICMPv6 juga menambahkan jenis dan kode baru yang diperlukan oleh fungsionalitas IPv6 yang baru. Ini termasuk jenis "Paket Terlalu Besar" dan kode kesalahan "opsi IPv6 yang tidak dikenali".

F. Manajemen Jaringan dan SNMP

Manajemen jaringan mencakup penyebaran, integrasi, dan koordinasi perangkat keras, perangkat lunak, dan elemen manusia untuk memonitor, menguji, polling, mengkonfigurasi, menganalisis, mengevaluasi, dan mengendalikan sumber daya jaringan dan elemen untuk memenuhi real-time, kinerja operasional, dan Persyaratan Kualitas Layanan dengan biaya yang masuk akal.

1. Kerangka Manajemen Jaringan



Gambar 5.8 Elemen manajemen jaringan: Mengelola server, Perangkat Terkelola, Data MIB, Agen Jarak Jauh, SNMP

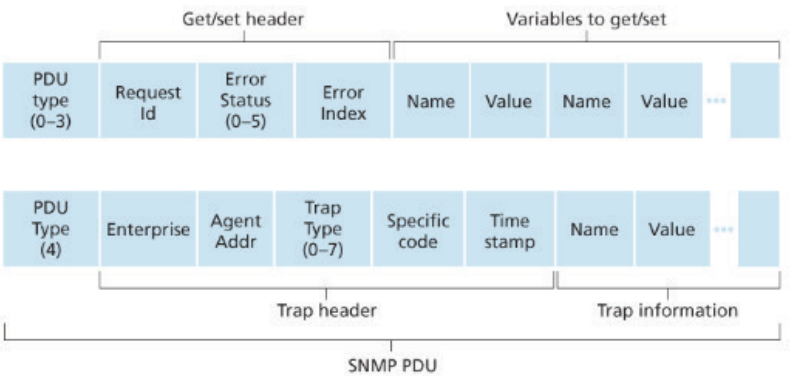
Komponen kunci dari manajemen jaringan:

- **Server pengelola** adalah lokus aktivitas untuk manajemen jaringan; itu mengontrol, pengumpulan, pemrosesan, analisis, dan/atau tampilan informasi manajemen jaringan.
- **Perangkat yang dikelola** yang mungkin dapat berupa host, router, switch, middlebox, modem, termometer, atau perangkat lain yang terhubung ke jaringan.
- ***Management Information Base* (IMB)** adalah pengumpulan setiap objek yang dikelola dalam perangkat yang dikelola terkait informasi. Objek MIB seperti jumlah datagram IP yang dibuang di router karena kesalahan dalam datagram IP header, atau jumlah segmen UDP yang diterima di sebuah host; objek MIB ditentukan dalam bahasa deskripsi data yang dikenal sebagai SMI (*Structure of Management Information*).
- **Agen manajemen jaringan** adalah sebuah proses yang berjalan di perangkat terkelola yang berkomunikasi dengan server pengelola, mengambil tindakan lokal di perangkat terkelola di bawah perintah dan kendali server pengelola.

› **Protokol manajemen jaringan** adalah protokol yang berjalan antara server pengelola dan perangkat yang dikelola, memungkinkan pengelolaan server untuk menanyakan status perangkat yang dikelola dan secara tidak langsung mengambil tindakan pada perangkat ini melalui agen. Protokol manajemen jaringan tidak mengelola jaringan itu sendiri, sebaliknya ia menyediakan kemampuan yang dapat digunakan oleh administrator jaringan untuk mengelola jaringan.

2. Protokol Manajemen Jaringan Sederhana (SNMP)

Protokol Manajemen Jaringan Sederhana versi 2 (SNMPv2) [RFC 3416] adalah protokol lapisan aplikasi yang digunakan untuk menyampaikan kontrol manajemen jaringan dan pesan informasi antara server pengelola dan agen yang menjalankan atas nama server pengelola tersebut. Penggunaan SNMP yang paling umum adalah dalam mode respons-permintaan di mana server pengelola SNMP mengirim permintaan ke agen SNMP, yang menerima permintaan, melakukan beberapa tindakan, dan mengirim balasan ke permintaan. Biasanya, permintaan akan digunakan untuk kueri (mengambil) atau mengubah (mengatur) nilai objek MIB yang terkait dengan perangkat yang dikelola. Penggunaan umum kedua SNMP adalah agar agen mengirim pesan yang tidak diminta, dikenal sebagai pesan jebakan, ke server pengelola. Pesan perangkap digunakan untuk memberi tahu server pengelola tentang situasi luar biasa (misalnya, antarmuka tautan naik atau turun) yang mengakibatkan perubahan pada nilai objek MIB.



Gambar 5.9 Format PDU SNMP

BAB 6
LAPISAN TAUTAN DAN LAN

A. Pengenalan Lapisan Tautan

Lapisan tautan data (*data link layer*) adalah lapisan kedua dari bawah dalam model OSI, yang dapat melakukan konversi frame-frame jaringan yang berisi data yang mendeteksi kesalahan dan pentransmisian ulang terhadap frame yang gagal. MAC address juga diimplementasikan di dalam lapisan ini. Selain itu, beberapa perangkat seperti Network Interface Card (NIC), switch layer 2 serta bridge jaringan juga beroperasi di sini.

Lapisan data-link menawarkan layanan pentransferan data melalui saluran fisik. Pentransferan data tersebut mungkin dapat diandalkan atau tidak: beberapa protokol lapisan data-link tidak mengimplementasikan fungsi *Acknowledgment* untuk sebuah frame yang sukses diterima, dan beberapa protokol bahkan tidak memiliki fitur pengecekan kesalahan transmisi (dengan menggunakan checksumming). Pada kasus-kasus tersebut, fitur-fitur acknowledgment dan pendeteksian kesalahan harus diimplementasikan pada lapisan yang lebih tinggi, seperti halnya protokol Transmission Control Protocol (TCP) (lapisan transport).

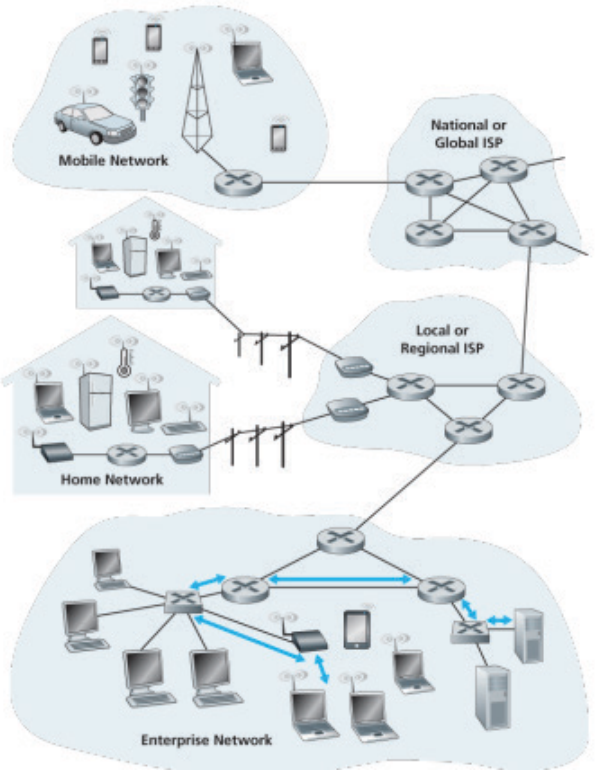
Tugas utama dari data link layer adalah sebagai fasilitas transmisi data mentah dan mentransformasi data tersebut ke saluran yang bebas dari kesalahan transmisi. Sebelum diteruskan ke *Network Layer*, lapisan data link melaksanakan tugas ini dengan memungkinkan pengirim memecah-mecah data input menjadi sejumlah data frame (biasanya berjumlah ratusan atau ribuan byte). Kemudian lapisan data link mentransmisikan

frame tersebut secara berurutan dan memproses *acknowledgement frame* yang dikirim kembali oleh penerima. Karena lapisan fisik menerima dan mengirim aliran bit tanpa mengindahkan arti atau arsitektur frame, maka tergantung pada lapisan data-link-lah untuk membuat dan mengenali batas-batas frame itu. Hal ini bisa dilakukan dengan cara membubuhkan bit khusus ke awal dan akhir frame.

Pendekatan yang umum dipakai adalah lapisan data link memecah aliran bit menjadi frame-frame dan menghitung nilai checksum untuk setiap frame-nya. Memecah-mecah aliran bit menjadi frame-frame lebih sulit dibandingkan dengan apa yang kita kira. Untuk memecah-mecah aliran bit ini, digunakanlah metode-metode khusus. Ada empat buah metode yang dipakai dalam pemecahan bit menjadi frame, yaitu:

- Karakter penghitung
- pemberian karakter awal dan akhir, dengan pengisian karakter
- Pemberian flag awal dan akhir, dengan pengisian bit
- Pelanggaran pengkodean Physical layer

Metode ini menggunakan sebuah field pada header untuk menspesifikasi jumlah karakter di dalam frame. Ketika data link layer pada komputer yang dituju melihat karakter penghitung, maka data link layer akan mengetahui jumlah karakter yang mengikutinya dan kemudian juga akan mengetahui posisi ujung framenya. Teknik ini bisa dilihat pada gambar 3 di bawah ini, dimana ada empat buah frame yang masing-masing berukuran 5,5,8 dan 8 karakter.



Gambar 6.1 Enam Lompatan Lapisan Tautan antara Host Nirkabel dan Server

1. Layanan yang Disediakan oleh Lapisan Tautan

Berikut layanan yang mungkin ditawarkan oleh protokol lapisan tautan:

› **Pembingkaian**. Untuk merangkum setiap datagram lapisan jaringan di dalam lapisan tautan bingkai sebelum transmisi melalui tautan. Bingkai terdiri dari bidang data, dimana lapisan jaringan datagram disisipkan, dan sejumlah bidang header. Struktur bingkai ditentukan oleh protokol lapisan tautan.

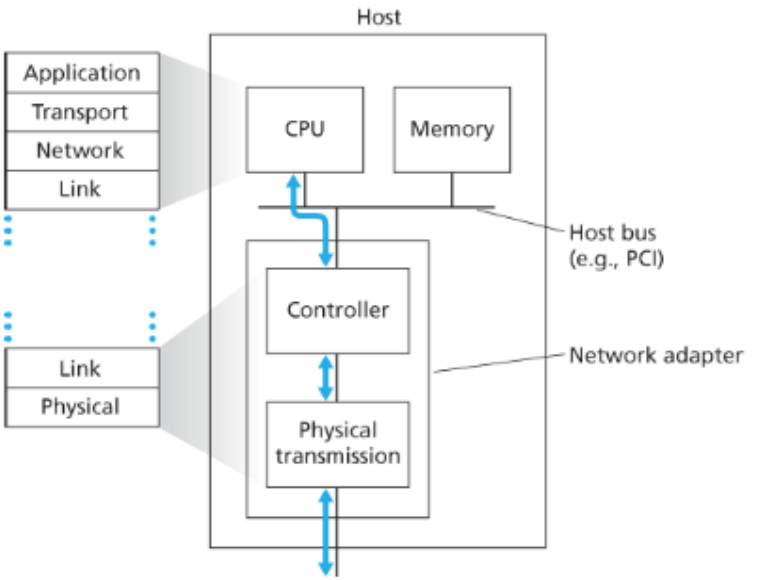
› **Akses tautan**. Sebuah *Medium Access Control* (MAC) menentukan aturan yang digunakan frame ditransmisikan ke tautan. Protokol MAC berfungsi mengoordinasikan transmisi bingkai dari banyak node.

› **Pengiriman yang andal**. Untuk menjamin pindahan setiap datagram lapisan jaringan melintasi tautan tanpa kesalahan.

› **Deteksi dan koreksi kesalahan**. Deteksi di lapisan tautan biasanya lebih canggih dan diterapkan di perangkat keras.

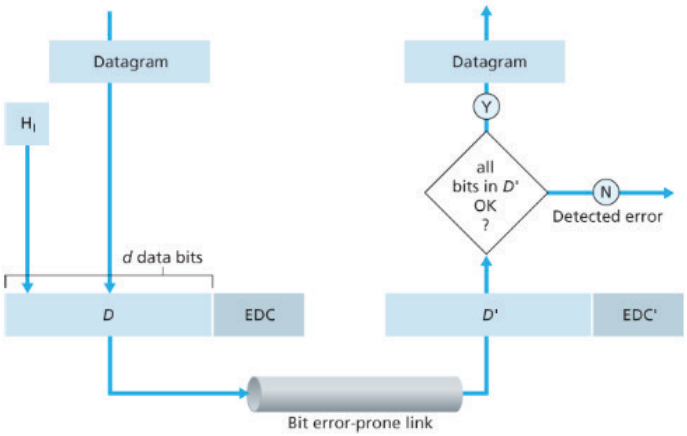
2. Tempat Implementasi Lapisan Tautan

Lapisan tautan diimplementasikan dalam kartu garis router. Untuk sebagian besar, lapisan tautan diimplementasikan dalam file adaptor jaringan, kadang-kadang juga dikenal sebagai *Network Interface Card* (NIC). Di jantung adaptor pengontrol lapisan tautan, biasanya chip tujuan khusus tunggal yang mengimplementasikan banyak hal layanan lapisan tautan.



Gambar 6.2 Adaptor jaringan: Hubungannya dengan komponen host lain dan dengan fungsionalitas tumpukan protokol

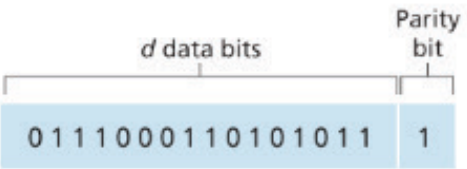
B. Teknis Deteksi Kesalahan dan Koreksi



Gambar 6.3 Deteksi Kesalahan dan Skenario Koreksi

1. Cek Paritas

Mungkin bentuk deteksi kesalahan yang paling sederhana adalah penggunaan bit paritas tunggal. Dalam skema paritas genap, pengirim hanya menyertakan satu bit tambahan dan memilih nilainya sedemikian sehingga jumlah total 1s dalam bit (informasi asli ditambah bit paritas) adalah genap. Untuk skema paritas ganjil, nilai bit paritas dipilih sedemikian sehingga ada jumlah ganjil 1s.



Gambar 6.4 Paritas Genap Satu Bit

Teknik ini akan mampu mendeteksi error bit yang jumlah bit errornya ganjil saja. Jika jumlah bit error jumlahnya genap maka teknik paritas bit akan mendeteksi tidak ada error padahal sebenarnya data yang dikirimkan terjadi error. Hanya sekitar 50% error yang dapat dideteksi dengan metode ini. Teknik ini juga dikenal kesederhanaannya dalam implementasi karena hanya menggunakan 1 bit saja untuk bit paritas (satu gate XOR untuk mengoperasikan paritas genap). Setiap error yang terdeteksi pada frame, maka pengirim harus mengirimkan

ulang frame tersebut karena teknik ini tidak bisa mengetahui posisi bit yang mengalami kesalahan.

2. Metode Pemeriksaan

Salah satu metode pemeriksaan sederhana adalah dengan menjumlahkan bilangan bulat ke bit ini dan menggunakan jumlah yang dihasilkan sebagai bit deteksi kesalahan. Metode pemeriksaan membutuhkan overhead paket yang relatif kecil. Karena deteksi kesalahan transport-layer diimplementasikan dalam perangkat lunak, penting untuk memiliki skema deteksi kesalahan sederhana dan cepat seperti checksumming. Di sisi lain, deteksi kesalahan pada lapisan tautan diimplementasikan dalam perangkat keras khusus dalam adaptor, yang dapat dengan cepat melakukan operasi CRC yang lebih kompleks.

3. Cycle Redudancy Check (CRC)

Teknik deteksi kesalahan yang digunakan secara luas di jaringan komputer saat ini didasarkan pada CRC kode. Kode CRC juga dikenal sebagai kode polinomial, karena dimungkinkan untuk melihat bit string untuk dikirim sebagai polinomial yang koefisiennya adalah nilai 0 dan 1 dalam string bit, dengan operasi pada string bit diartikan sebagai aritmatika polinomial.

Cyclic Redudancy Check (CRC) adalah kode pendeteksi kesalahan yang biasa digunakan dari jaringan digital dan perangkat penyimpanan untuk mendeteksi perubahan yang tidak disengaja pada data mentah. Blok data yang memasuki sistem ini mendapatkan nilai cek singkat yang dilampirkan, berdasarkan sisa pembagian polinomial isinya. Saat pengambilan, kalkulasi diulangi dan, jika nilai cek tidak cocok, tindakan korektif dapat diambil terhadap kerusakan data. CRC dapat digunakan untuk koreksi kesalahan (lihat filter bit).

CRC disebut demikian karena nilai *check* (verifikasi data) adalah redundansi (memperluas pesan tanpa menambahkan informasi) dan algoritme didasarkan pada kode *siklik*. CRC sangat populer karena mudah diimplementasikan dalam perangkat keras biner, mudah dianalisis secara matematis, dan sangat baik dalam mendeteksi kesalahan umum yang disebabkan oleh noise di saluran transmisi. Karena nilai cek memiliki panjang tetap, fungsi yang menghasilkannya terkadang digunakan sebagai fungsi hash.

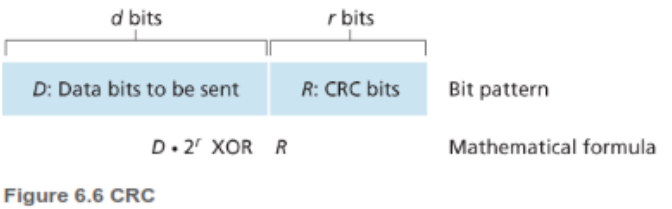


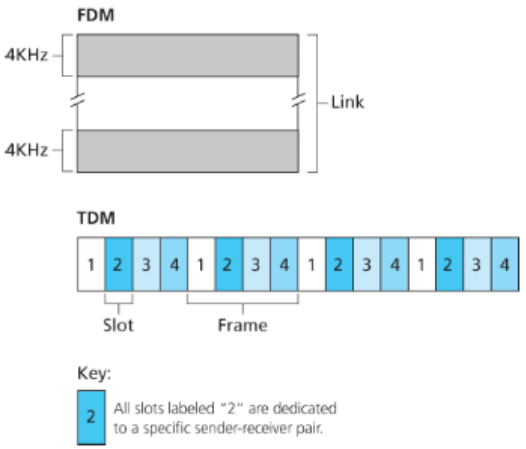
Figure 6.6 CRC

Gambar 6.5 CRC

C. Beberapa Link dan Akses Protokol

1. Protokol Partisi Saluran

Time-Division Multiplexing (TDM) dan *Frequency-Division Multiplexing* (FDM) adalah dua teknik yang bisa digunakan untuk mempartisi bandwidth saluran siaran di antara semua node yang berbagi saluran.



Gambar 6.6 Contoh TDM dan FDM empat node

TDM (*Time Division Multiplexing*) membagi waktu menjadi keraangka waktu dan selanjutnya membagi setiap kerangka waktu menjadi slot waktu N. Setiap slot waktu kemudian ditugaskan ke salah satu dari N node. Sementara TDM dan FDM masing-masing menetapkan slot waktu dan frekuensi, untuk node. CDMA memberikan kode yang berbeda untuk setiap node. Setiap node kemudian menggunakan kode uniknya untuk menyandingkan bit

data yang dikirimkannya. Jika kode dipilih dengan hati-hati, jaringan CDMA memiliki properti luar biasa yang dapat ditransmisikan secara berbeda oleh node yang berbeda secara simultan namun masing-masing penerima menerima bit data yang dikodekan meskipun ada gangguan pada transmisi oleh node lain.

2. Protokol Akses Acak

Dalam protokol akses acak, node pengirim selalu mentransmisikan pada tingkat penuh saluran, yaitu, R bps. Ketika ada tabrakan, setiap node yang terlibat dalam tabrakan berulang kali mentransmisikan ulang frame-nya (yaitu paket) sampai frame-nya melewati tanpa tabrakan. Tetapi ketika sebuah node mengalami tabrakan, itu tidak harus mentransmisikan ulang frame segera. Alih-alih itu menunggu penundaan acak sebelum mengirimkan kembali frame. Setiap node yang terlibat dalam tabrakan memilih penundaan acak independen. Karena penundaan acak dipilih secara independen, ada kemungkinan bahwa salah satu node akan memilih penundaan yang cukup kurang dari keterlambatan node bertabrakan lainnya dan karena itu akan dapat menyelinap bingkai ke dalam saluran tanpa tabrakan.

› *Slot ALOHA*

Protokol akses acak paling sederhana. Slot ALOHA diasumsikan sebagai berikut:

1) Semua frame terdiri dari bit L persis.
2) Waktu dibagi menjadi slot-slot ukuran L/R detik (satu slot = waktu mentransmisikan satu frame).
3) Node mulai mengirim frame hanya pada awal slot
4) Node disinkronkan sehingga setiap node tahu kapan slot dimulai.
5) Jika dua atau lebih frame bertabrakan dalam slot, maka semua node mendeteksi peristiwata tabrakan sebelum slot berakhir.

› *Carrier Sense Multiple Access* (CSMA)

Carrier Sense Multiple Access with Collision Detection atau sering disingkat menjadi CSMA/CD adalah sebuah metode *media access control* (MAC) yang digunakan oleh teknologi jaringan

Eternet. Dengan metode ini, sebuah node jaringan yang akan mengirim data ke node tujuan pertama-tama akan memastikan bahwa jaringan sedang tidak dipakai untuk transfer dari dan oleh node lainnya. Jika pada tahap pengecekan ditemukan transmisi data lain dan terjadi tabrakan (*collision*), maka node tersebut diharuskan mengulang permohonan (*request*) pengiriman pada selang waktu berikutnya yang dilakukan secara acak (*random*). Dengan demikian maka jaringan efektif bisa digunakan secara bergantian.

3. Protokol Taking-Turns

Dua sifat yang diinginkan dari protokol akses berganda adalah (1) ketika hanya satu node aktif, node aktif memiliki throughput R bps, dan (2) ketika M node aktif, maka setiap node aktif memiliki throughput hampir R / M bps. Protokol ALOHA dan CSMA memiliki properti pertama ini tetapi bukan yang kedua.

Kita akan membahas dua protokol penting disini:

a. Protokol pemungutan suara yang membutuhkan salah satu simpul yang ditetapkan sebagai node master, protokol ini menghilangkan tabrakan dan slot kosong yang mengganggu protokol akses acak.
b. Protokol token-passing (pengambilan giliran) yang memiliki node master.

4. DOCSIS (Data Over-Cable Service Interface Specification)

DOCSIS menentukan jaringan kabel data arsitektur dan protokolnya. DOCSIS menggunakan FDM untuk membagi downstream (CMTS ke modem) dan segmen jaringan hulu (modem ke CMTS) menjadi beberapa saluran frekuensi. Setiap hilir saluran selebar 6 MHz, dengan throughput maksimum sekitar 40 Mbps per saluran; setiap saluran hulu memiliki maksimum lebar saluran 6,4 MHz, dan throughput hulu maksimum sekitar 30 Mbps. Setiap hulu dan saluran hilir adalah saluran siaran.

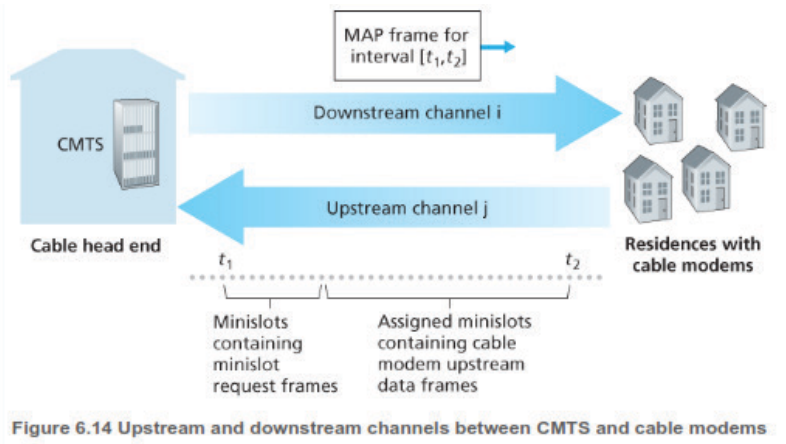


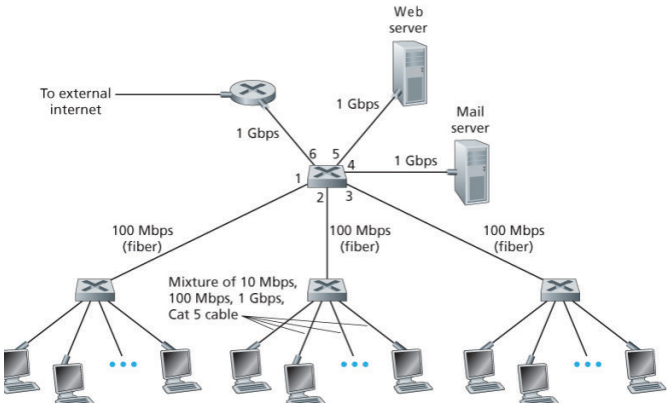
Figure 6.14 Upstream and downstream channels between CMTS and cable modems

Gambar 6.7 *Upstream* dan *Downstream* antara CMTS dan Modem Kabel

D. Switched Local Area Network (LAN)

LAN adalah kependekan dari Local Area Network yang merupakan suatu jaringan yang di mana perangkat keras dan perangkat lunak bisa saling berkomunikasi dalam daerah yang terbatas. LAN hanya bisa menjangkau daerah yang sangat terbatas. misalnya hanya dapat menjangkau dalam satu gedung saja.

Local Area Network atau jaringan komputer lokal adalah sebuah jaringan komputer yang terbatas hanya pada sebuah wilayah kecil saja. Dari pengertian di atas dapat diambil kesimpulan bahwa LAN ini hanya terbatas pada suatu wilayah/ kompleks saja. Contoh dari LAN yang sering kita temui yaitu jaringan komputer di kompleks gedung perkantoran, warnet, cafe rumah pribadi dll.

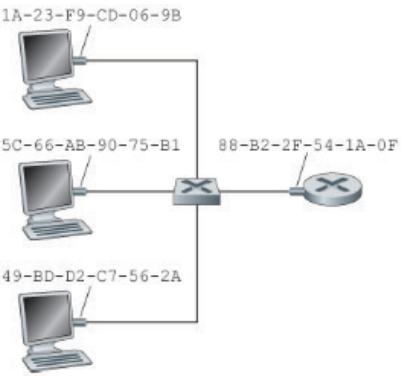


Gambar 6.8 Jaringan Kelembagaan yang dihubungkan bersama oleh Empat Sakelar

1. Pengenalan Lapisan Tautan dan ARP

› MAC Address

MAC Address (Media Access Control address) adalah alamat fisik suatu interface jaringan (seperti ethernet card pada komputer, interface/port pada router, dan node jaringan lain) yang bersifat unik dan berfungsi sebagai identitas perangkat tersebut. Secara umum MAC Address dibuat dan diberikan oleh pabrik pembuat NIC (Network Interface Card) dan disimpan secara permanen pada ROM (Read Only Memory) perangkat tersebut. MAC address juga biasa disebut Ethernet Hardware Address (EHA), Hardware Addres, atau Physical Address.

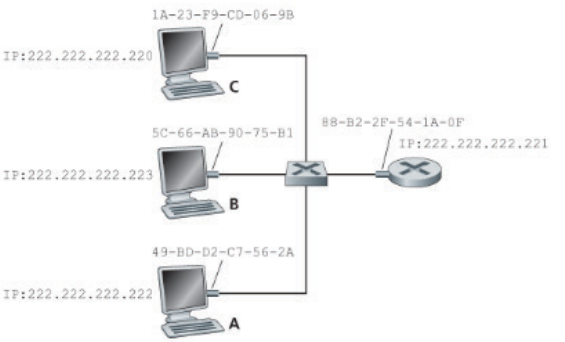


Gambar 6.9 Setiap antarmuka yang terhubung ke LAN memiliki alamat MAC yang unik

Alamat MAC adaptor mrmiliki struktur datar dan tidak berubah dimana pun adaptor itu pergi, sedangkan alamat IP memiliki struktur hierarki dan kebutuhan alamat IP host untuk diubah ketika host berpindah, yaitu mengubah jaringan tempat ia terhubung.

› *Address Resolution Protocol* (ARP)

Protokol ARP atau Address Resolution Protocol merupakan sebuah protokol yang bertanggung jawab mencari tahu Mac Address atau alamat hardware dari suatu Host yang tergabung dalam sebuah jaringan LAN dengan memanfaatkan atau berdasarkan IP Address yang terkonfigurasi pada Host yang bersangkutan.



Gambar 6.10 Setiap antarmuka pada LAN memiliki alamat IP dan alamat MAC

2. Ethernet

Semua teknologi Ethernet menyediakan layanan tanpa koneksi ke lapisan jaringan. Beberapa alasan kenapa menggunakan Etherner:

a. Ethernet adalah LAN berkecepatan tinggi pertama yang digunakan secara luas

b. Token ring, FDDI, dan ATM lebih kompleks dan mahal daripada Ethernet, yang selanjutnya membuat administrator jaringan tidak mau beralih.

c. Alasan paling kuat untuk beralih ke teknologi LAN lan biasanya adalah tingkat data yang lebih tinggi dari teknologi baru; namun ethernet menghasilkan versi yang beroperasi pada kecepatan data yang sama atau lebih tinggi.

Struktur Frame Ethernet

Gambar 6.11 Struktur Bingkai *Ethernet*

Berikut enam bidang bingkai Ethernet:

a. **Bidang Data (46 – 1.500 byte)**. Bidang ini membawa datagram IP. *Maximum Transmittion Unit* (MTU) Ethernet adalah 1.500 byte, jika lebih dari itu maka host harus memecah datagram. Ukuran minimum data bidang adalah 46 byte, jika kurang dari itu maka field harus 'diisi' untuk hingga 46 byte.

b. **Alamat tujuan (6 byte)**. Kolom ini berisi alamat MAC dari adaptor tujuan, BB-BB-BB-BB-BB-BB.

c. **Alamat sumber (6 byte)**. Bidang ini berisi alamat MAC dari adaptor yang mentransmisikan bingkai ke LAN, contoh AA-AA-AA-AA-AA-AA.

d. **Ketik kolom (2 byte)**. Bidang tipe ini mengizinkan Ethernet untuk membuat protokol lapisan jaringan multipleks.

e. ***Cyclic redudancy check* (CRC) (4 byte)**. Tujuan CRC bidang adalah untuk memungkinkan adaptor penerima untuk mendeteksi kesalahan bit dalam bingkai.

f. **Pembukaan (8 byte)**. Masing-masing dari 7 byte yang pertama dari pembukaan memiliki nilai 10101010 yang berfungsi untuk 'membangunkan' adaptor penerima dan untuk menyinkronkan jam mereka dengan yang ada di jam pengirim; byte terakhir adalah 10101011 yang berfungsi untuk memperingatkan adaptor B bahwa 'hal penting' akan segera datang.

3. Sakelar Lapisan Tautan

Berikut akan kita bahas bagaimana sakelar beroperasi

› Meneruskan dan Memfilter

Pemfilteran adalah fungsi sakelar yang menentukan apakah bingkai harus diteruskan ke beberapa antarmuka atau harus dijatuhkan. **Penerusan** adalah fungsi sakelar yang menentukan antarmuka ke mana bingkai harus diarahkan, dan kemudian

memindahkan bingkai ke antarmuka tersebut. Pemfilteran dan penerusan dilakukan dengan tabel sakelar yang berisi (1) Alamat MAC, (2) sakelar antarmuka yang mengarah ke alamat, dan (3) waktu entri ditempatkan di file meja. Tabel switch berisi entri untuk beberapa, tapi tidak harus semua, dari host dan router di LAN.

› Self-Learning

Sebuah saklar memiliki properti yang luar itu tabelnya dibuat secara otomatis, dinamis, dan otonom — tanpa intervensi apa pun dari jaringan administrator atau dari protokol konfigurasi. Dengan kata lain, sakelar adalah pembelajaran mandiri. Kemampuan ini dicapai sebagai berikut:

1) Tabel sakelar awalnya kosong.
2) Untuk setiap frame masuk yang diterima pada sebuah interface, switch menyimpan MAC dalam tabelnya (1) alamat di bidang alamat sumber bingkai, (2) antarmuka dari mana bingkai datang, dan (3) waktu saat ini.
3) Switch menghapus alamat dalam tabel jika tidak ada frame yang diterima dengan alamat itu sebagai alamat sumber setelah beberapa periode waktu (waktu penuaan).

› Properti dari Link-Layer Switching

Beberapa keuntungan menggunakan sakelar, daripada tautan siaran seperti topologi bintang berbasis bus atau hub:

1) **Penghapusan tabrakan**. Switch frame buffer dan tidak pernah mengirimkan lebih dari satu frame segmen pada satu waktu. Seperti halnya router, throughput agregat maksimum dari sakelar adalah jumlah dari semua tingkat antarmuka sakelar. Dengan demikian, sakelar memberikan peningkatan kinerja yang signifikan melalui LAN dengan tautan siaran.
2) **Tautan heterogen**. Karena sakelar mengisolasi satu tautan dari tautan lain, tautan berbeda di LAN dapat beroperasi pada kecepatan yang berbeda dan dapat berjalan melalui media yang berbeda. Jadi, sakelar sangat ideal untuk mencampur warisan peralatan dengan yang baru peralatan.

3) **Pengelolaan**. Switch juga memudahkan manajemen jaringan. Misalnya, jika adaptor rusak dan terus-menerus mengirimkan bingkai Ethernet (disebut adaptor jabbering), sakelar dapat mendeteksi masalah dan secara internal lepaskan adaptor yang rusak tempat tidur dan berkendara kembali untuk bekerja untuk memperbaiki masalah.

4. Virtual Local Area Network (VLAN)

LAN institusional modern sering kali dikonfigurasi secara hierarki, dengan setiap grup kerja (departemen) memiliki LAN yang dialihkan sendiri yang terhubung ke LAN yang dialihkan dari grup lain melalui hierarki sakelar. Sementara konfigurasi seperti itu bekerja dengan baik secara ideal dunia, dunia nyata seringkali jauh dari ideal. Tiga kelemahan dapat diidentifikasi dalam konfigurasi :

› Kurangnya isolasi lalu lintas.

› Penggunaan sakelar yang tidak efisien.

› Mengelola pengguna.

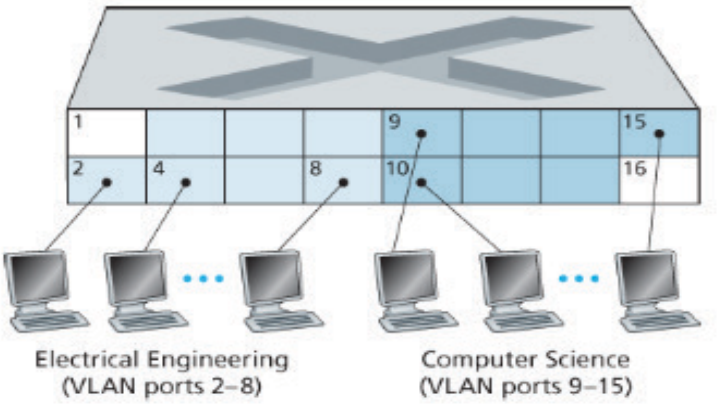


Figure 6.25 A single switch with two configured VLANs

Gambar 6.12 Sakelar tunggal dengan dua VLAN yang dikonfigurasi

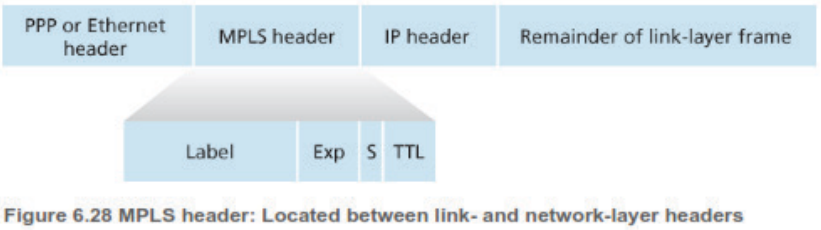
E. Virtualisasi Tautan: Jaringan Sebagai Tautan Lapisan

Internet memvirtualisasikan jaringan telepon, memandang jaringan telepon, memandang jaringan telepon sebagai lapisan sambungan teknologi yang menyediakan konektivitas lapisan tautan antara dua host internet.

Kita akan membahas jaringan *Multi Protocol Label Switching* (MPLS). MPLS adalah *packet-switched*, jaringan sirkuit virtual dengan sendirinya. Yang berarti memiliki miliknya sendiri format paket dan perilaku penerusan

1. Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching (MPLS) adalah teknologi penyampaian paket pada jaringan backbone berkecepatan tinggi. Asas kerjanya menggabungkan beberapa kelebihan dari sistem komunikasi circuit-switched dan packet-switched yang melahirkan teknologi yang lebih baik dari keduanya. Sebelumnya, paket-paket diteruskan dengan protokol routing seperti OSPF, IS-IS, BGP, atau EGP. Protokol routing berada pada lapisan network (ketiga) dalam sistem OSI, sedangkan MPLS berada di antara lapisan kedua dan ketiga.



Gambar 6.13 MPLS header: Terletak di antara *Link* dan *Network-Layer* Header

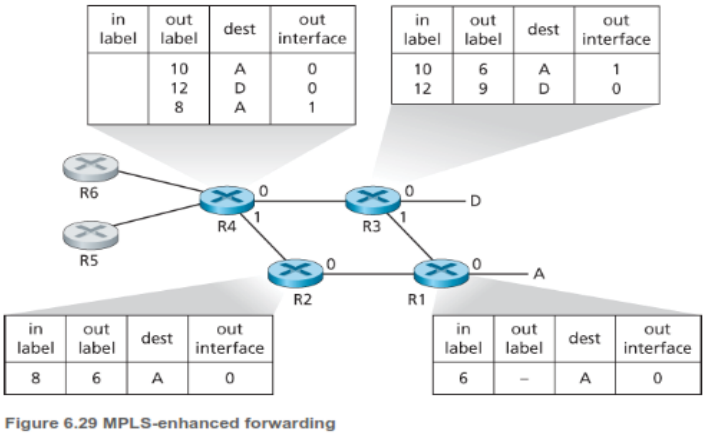


Figure 6.29 MPLS-enhanced forwarding

Gambar 6.14 Penerusan yang ditingkatkan MPLS

F. Jaringan Pusat Data (DCN)

Pusat Data adalah kumpulan sumber daya yang saling berhubungan menggunakan jaringan komunikasi, DCN memegang peran penting dalam pusat data, karena menghubungkan semua sumber daya pusat data bersama-sama. DCN harus dapat diskalakan dan efisien untuk menghubungkan puluhan atau bahkan ratusan ribu server untuk menangani tautan komputasi Cloud yang terus meningkat. Pusat data saat ini dibatasi oleh jaringan interkoneksi.

Jaringan Pusat Data mendukung dua jenis lalu lintas: lalu lintas yang mengalir antara klien eksternal dan internal host dan lalu lintas yang mengalir di antara host internal.

- Penyeimbangan Beban

  Pusat data yang besar sering kali memiliki beberapa penyeimbang beban, masing-masing dikhususkan untuk sekumpulan aplikasi cloud tertentu. Penyeimbang beban seperti itu disebut sebagai "Layer-4 Switch" karena membuat keputusan berdasarkan nomor port tujuan serta alamat IP tujuan dalam paket.

- Arsitektur Hierarkis

  Untuk pusat data kecil yang menampung beberapa ribu host, jaringan sederhana yang terdiri dari perbatasan router, penyeimbang beban,

dan beberapa puluhan rak semuanya dapat dihubungkan oleh satu sakelar Ethernrt mungkin cukup. Tetapi untuk menskalakan hingga puluhan hingga ratusan ribu host, pusat data sering kali menggunakan file hierarki router dan switch.

- Tren dalam Jaringan Pusat Data

  Salah satu tren jaringan pusat data yang digunakan oleh raksasa awan internet seperti Google, Facebook, Amazon dan Microsoft terus berlanjut menyebarkan arsitektur interkoneksi baru dan protokol jaringan yang mengatasi kelemahan dari desain hirarkis tradisional. Salah satu pendekatan tersebut adalah untuk menggantikan hirarki switch dan router dengan sebuah topologi yang terhubung sepenuhnya

BAB 7
JARINGAN NIRKABEL DAN SELULER

A. Pengenalan

Berikut elemen yang terdapat dalam jaringan nirkabel:

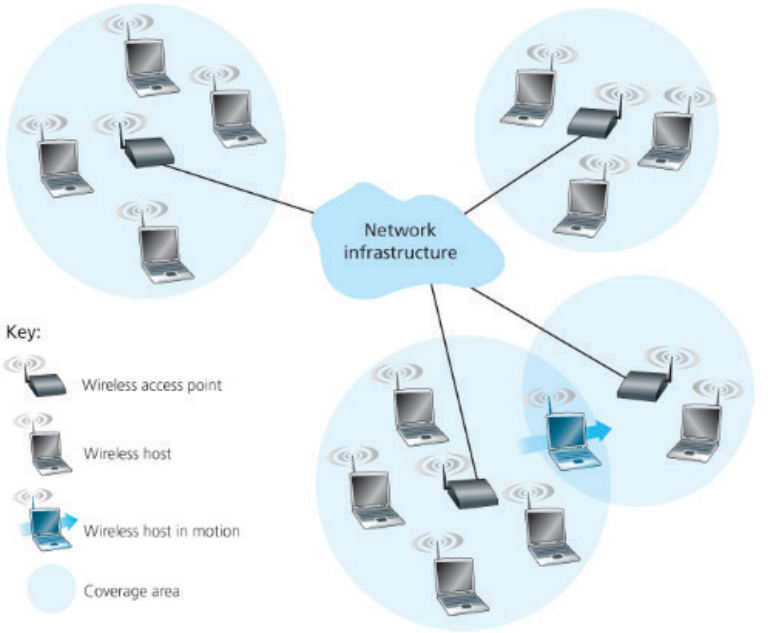


Figure 7.1 Elements of a wireless network

Gambar 7.1 Elemen jaringan nirkabel

- **Host Nirkabel.** Host adalah perangkat sistem akhir yang menjalankan aplikasi. Host nirkabel bisa berupa laptop, tablet, smartphone, atau komputer dekstop.
- **Tautan nirkabel.** Tautan terhubung ke *base* station atau ke host nirkabel lain melalui tautan komunikasi nirkabel. Teknologi tautan nirkabel yang berbeda memiliki kecepatan transmisi yang berbeda dan dapat mengirimkan melalui jarak yang berbeda.
- **Infrastruktur jaringan.** Adalah jaringan yang lebih besar yang mungkin diinginkan oleh host nirkabel menyampaikan.
- **Stasiun pangkalan.** Base station adalah bagian penting dari infrastruktur jaringan nirkabel. tidak seperti host nirkabel dan tautan nirkabel. Base stasiun bertanggung jawab untuk mengirim dan menerima data dan dari host nirkabel itu dikaitkan dengan stasiun pangkalan itu. BTS sering kali akan bertanggung jawab untuk mengoordinasikan transmisi beberapa host nirkabel yang terkait dengannya. Menara seluler di jaringan seluler dan titik akses di nirkabel 802.11LAN adalah contoh stasiun basis Infrastruktur jaringan. Ini adalah jaringan yang lebih besar yang diinginkan oleh host nirkabel. Infrastruktur seperti stasiun pangkalan dalam jaringan sebagai berikut:
    - **Single-hop**, **berbasis infrastruktur**. Jaringan ini memiliki stasiun pangkalan yang terhubung ke sebuah jaringan kabel yang lebih besar (mis., Internet). Selain itu, semua komunikasi antara stasiun pangkalan ini dan host nirkabel melalui satu hop nirkabel. Jaringan 802.11 yang Anda gunakan di ruang kelas, kafe, atau perpustakaan; dan jaringan data 4G LTE semuanya termasuk dalam kategori ini.
    - **Single-hop, tanpa infrastruktur**. Dalam jaringan ini, tidak ada stasiun pangkalan yang terhubung ke a jaringan nirkabel. Jaringan Bluetooth dan jaringan 802.11 dalam mode ad hoc adalah jaringan single-hop, tanpa infrastruktur.
    - **Multi-hop, berbasis infrastruktur**. Dalam jaringan ini, ada stasiun pangkalan yang ditransfer kejaringan yang lebih besar. Namun, beberapa node nirkabel mungkin harus menyampaikan komunikasinya melalui yang lain node nirkabel untuk

berkomunikasi melalui stasiun pangkalan. Beberapa jaringan sensor nirkabel dan apa yang disebut jaringan mesh nirkabel.

› **Multi-hop, tanpa infrastruktur**. Tidak ada stasiun pangkalan di jaringan ini, dan node mungkin harus menyampaikan pesan di antara beberapa node lain untuk mencapai tujuan. Node juga mungkin mobile, dengan konektivitas berubah di antara node kelas jaringan yang dikenal sebagai mobile ad hoc jaringan (MANET). Manet merupakan kumpulan node bergerak secara dinamis yang mampu membentuk jaringan sementara tanpa memerlukan infrastruktur yang telah ada[Fauzan, 2014]. Jika mobile node adalah kendaraan, jaringan adalah jaringan ad hoc kendaraan (VANET).

B. Tautan Nirkabel dan Karakteristik Jaringan

Perbedaan antara tautan kabel dan tautan nirkabel :

- Mengurangi kekuatan sinyal, radiasi elektromagnetik melemahkan saat ia melewati materi.
- Gangguan dari sumber radio lain yang mentransmisikan dalam pita frekuensi yang sama akan saling mengganggu.
- Multipath propagation. Perambatan multipath terjadi ketika bagian dari gelombang elektromagnetik memantulkan benda dan tanah, mengambil jalur dengan panjang berbeda antara pengirim dan penerima.

Beberapa karakteristik lapisan fisik yang penting dalam memahami higher-layer wireless Communication :

- Untuk skema modulasi tertentu, semakin tinggi SNR, semakin rendah BER.
- Untuk SNR yang diberikan, teknik modulasi dengan laju transmisi bit yang lebih tinggi (baik dalam kesalahan atau tidak) akan memiliki BER yang lebih tinggi.
- Pemilihan dinamis dari teknik modulasi lapisan fisik dapat digunakan untuk mengadaptasi teknik modulasi untuk kondisi saluran.

1. Code Division Multiple Access (CDMA)

CDMA adalah sebuah bentuk pemultipleksan (bukan skema pemodulasian) dan sebuah metode akses secara bersama yang membagi kanal tidak berdasarkan waktu atau frekuensi, tetapi dengan cara mengkodekan data dengan sebuah kode khusus yang diasosiasikan dengan tiap kanal yang ada dan menggunakan sifat-sifat interferensi kontruktif dari kode-kode khusus itu untuk melakukan pemultipleksan. CDMA sangat penting di dunia nirkabel. Dalam protokol CDMA, setiap bit yang dikirim dikodekan dengan mengalikan bit dengan sinyal (kode) itu perubahan pada kecepatan yang jauh lebih cepat (dikenal sebagai chipping rate) daripada urutan asli bit data.

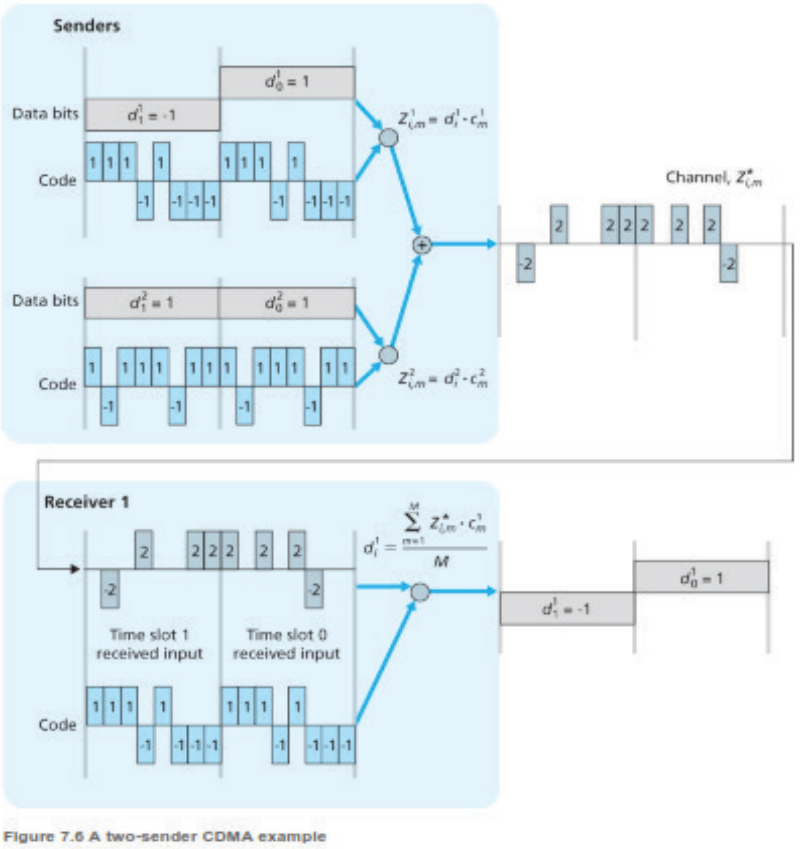


Figure 7.6 A two-sender CDMA example

Gambar 7.2 Contoh CDMA dua pengirim

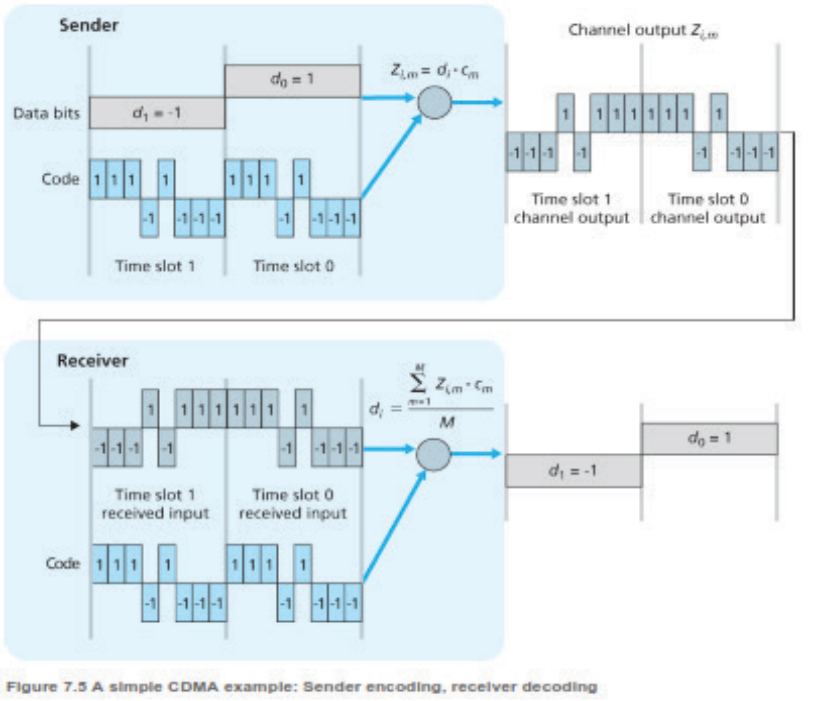


Figure 7.5 A simple CDMA example: Sender encoding, receiver decoding

Gambar 7.3 contoh CDMA sederhana: Pengkodean pengirim, decoding penerima

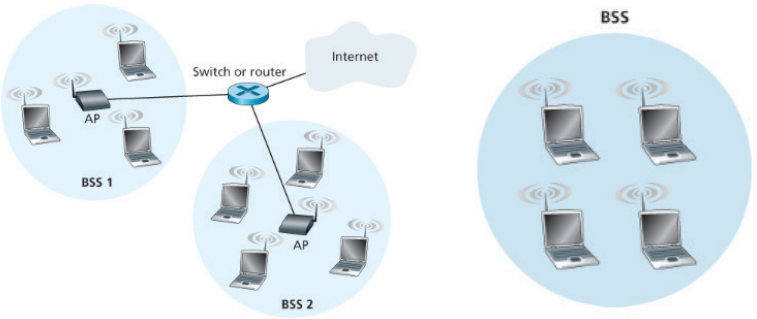
C. WiFi: 802.11 Wireless LANs

Meresap di tempat kerja, rumah, lembaga pendidikan, kafe, bandara, dan sudut jalan, LAN nirkabel kini menjadi salah satu teknologi jaringan akses paling penting di Internet saat ini. Meskipun banyak teknologi dan standar untuk LAN nirkabel dikembangkan pada 1990-an, satu kelas standar tertentu jelas telah muncul sebagai pemenang: LAN nirkabel IEEE 802.11, juga dikenal sebagai WiFi.

1. The 802.11 Architecture

Arsitektur 802.11 adalah set layanan dasar (BSS). BSS berisi satu atau lebih stasiun nirkabel dan stasiun basis pusat, yang dikenal sebagai titik akses di 802.11 bahasa. Dalam jaringan rumah yang khas, ada satu AP dan satu router yang menghubungkan BSS ke Internet. Seperti halnya perangkat Ethernet, setiap stasiun nirkabel 802.11 memiliki

alamat MAC 6-byte yang disimpan dalam firmware adaptor stasiun. Standar 802.11 tidak menentukan algoritma untuk memilih AP mana yang tersedia untuk dikaitkan dengan algoritma yang diserahkan kepada perancang firmware dan perangkat lunak 802.11 di nirkabel mesin. Meskipun kekuatan sinyal tinggi baik, kekuatan sinyal bukan satu-satunya titik Akses karakteristik yang akan menentukan kinerja yang diterima perangkat. AP yang dipilih mungkin memiliki sinyal yang kuat,tetapi mungkin kelebihan beban dengan perangkat afiliasi lainnya, sementara AP yang dibongkar tidak dipilih karena sedikit sinyal lemah.



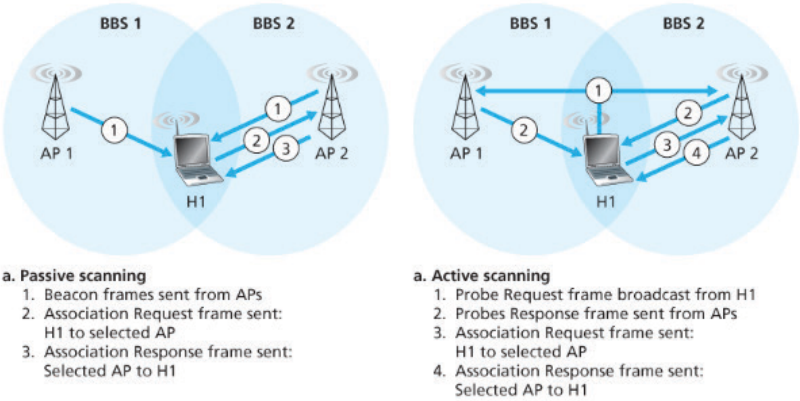
Gambar 7.4 Arsitektur IEEE 802.11 LAN dan An IEEE 802.11 ad hoc network

Channels and Association

Setiap stasiun nirkabel perlu dikaitkan dengan AP sebelum dapat mengirim atau menerima data pemain jaringan. Ketika seorang administrator jaringan menginstal suatu AP, administrator tersebut menetapkan Service Set satu atau dua kata pengenal (SSID) ke titik akses. Administrator juga harus menetapkan nomor saluran ke AP. Untuk memahami nomor saluran. Hutan WiFi adalah setiap lokasi fisik di mana sebuah stasiun nirkabel menerima sinyal yang cukup kuat dari dua atau lebih AP. Untuk membuat hubungan dengan AP tertentu, perangkat nirkabel mungkin diperlukan untuk mengotentikasi ke AP. 802.11 LAN nirkabel menyediakan sejumlah alternatif untuk otentikasi dan akses. Salah satu pendekatan, yang digunakan oleh banyak perusahaan, mengizinkan akses ke jaringan nirkabel berbasis pada alamat MAC perangkat.

Standar 802.11 tidak menentukan algoritme untuk memilih AP mana yang tersedia untuk dikaitkan dengan algoritme tersebut

diserahkan kepada perancang firmware dan perangkat lunak 802.11 di nirkabel Anda. Biasanya, perangkat memilih AP yang bingkai suarnya diterima dengan sinyal tertinggi kekuatan. Kekuatan sinyal bukanlah satu-satunya AP karakteristik yang akan menentukan kinerja yang diterima perangkat.



Gambar 7.5 Pemindaian aktif dan pasif untuk titik akses

2. The 802.11 MAC Protocol

Setelah perangkat nirkabel dikaitkan dengan AP, ia dapat mulai mengirim dan menerima bingkai data dan dari titik akses. Tetapi karena beberapa perangkat nirkabel, atau AP itu sendiri mungkin ingin mengirimkan data frame pada saat yang sama melalui saluran yang sama, diperlukan protokol akses ganda untuk mengoordinasikan transmisi. Protokol akses acak adalah disebut sebagai CSMA dengan menghindari tabrakan, atau lebih ringkasnya sebagai CSMA / CA. Meskipun kedua Ethernet dan 802.11 menggunakan akses acak pembawa-sensing, dua MAC protokol memiliki perbedaan penting. Dealing with Hidden Terminals: RTS and CTS mengapa terminal tersembunyi bisa bermasalah. Misalkan Station H1 mentransmisikan a bingkai dan setengah jalan melalui transmisi H1, Station H2 ingin mengirim bingkai ke AP. H2, tidak mendengar transmisi dari H1. pertama akan menunggu interval DIFS dan kemudian mengirimkan frame, menghasilkan tabrakan. Saluran akan terbuang sia-sia selama periode transmisi H1, juga selama transmisi H2. Untuk menghindari masalah ini, protokol IEEE 802.11 memungkinkan stasiun untuk menggunakan Permintaan singkat kirim bingkai kontrol (RTS) dan

bingkai kontrol Hapus untuk Mengirim (CTS) singkat untuk memesan akses ke saluran. Ketika pengirim ingin mengirim DATA.

Penggunaan bingkai RTS dan CTS dapat meningkatkan kinerja dalam dua cara penting:

a. Masalah stasiun tersembunyi dikurangi, karena bingkai DATA panjang ditransmisikan hanya setelah saluran telah dipesan.
b. Karena frame RTS dan CTS pendek, tabrakan yang melibatkan frame RTS atau CTS hanya akan bertahan lama untuk durasi frame RTS atau CTS pendek. Setelah frame RTS dan CTS benar ditransmisikan, frame DATA dan ACK berikut harus ditransmisikan tanpa tabrakan.

3. The IEEE 802.11 Frame

Meskipun frame 802.11 memiliki banyak kesamaan dengan frame Ethernet, frame ini juga mengandung sejumlah bidang yang khusus untuk penggunaannya untuk tautan nirkabel. Payload and CRC Fields:Inti dari frame adalah payload, yang biasanya terdiri dari datagram IP atau paket ARP. Address Fields untuk memindahkan datagram lapisan jaringan dari nirkabel stasiun melalui AP ke antarmuka router. Bidang alamat keempat digunakan saat AP meneruskan frame satu sama lain dalam mode ad hoc.

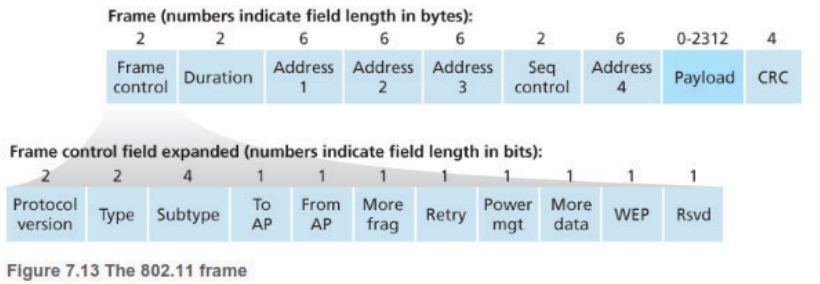


Figure 7.13 The 802.11 frame

Gambar 7.6 Frame 802.11

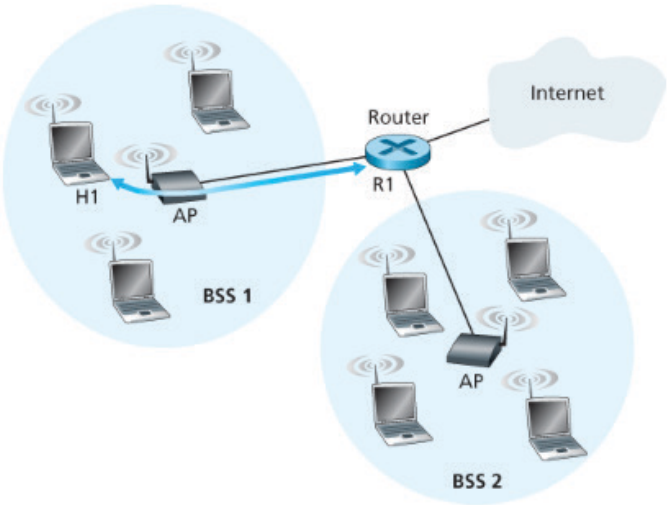
Standar 802.11 mendefinisikan bidang-bidang ini sebagai berikut:

› Address 2 adalah alamat MAC dari stasiun yang mentransmisikan frame. Jadi, jika stasiun nirkabel mentransmisikan frame, alamat MAC stasiun itu dimasukkan dalam bidang address 2. Begitu pula

jika AP mentransmisikan frame, alamat MAC AP dimasukkan di bidang alamat 2.

› Address 1 adalah alamat MAC dari stasiun nirkabel yang menerima frame. Jadi kalau mobile stasiun nirkabel mentransmisikan frame, alamat 1 berisi alamat MAC dari AP tujuan. Demikian pula, jika AP mentransmisikan frame, alamat 1 berisi alamat MAC tujuan stasiun nirkabel

› .Address 3 BSS (terdiri dari AP dan stasiun nirkabel) adalah bagian dari subnet, dan bahwa subnet ini terhubung ke subnet lain melalui beberapa antarmuka router. Alamat 3 berisi alamat MAC dari antarmuka router.



Gambar 7.7 Penggunaan kolom alamat pada frame 802.11: Pengiriman frame antara H1 dan R1

4. Mobility in the Same IP Subnet

Untuk meningkatkan jangkauan fisik LAN nirkabel, perusahaan dan universitas akan sering menggunakan beberapa BSS dalam subnet IP yang sama. Ini secara alami memunculkan masalah mobilitas di antara BSS.

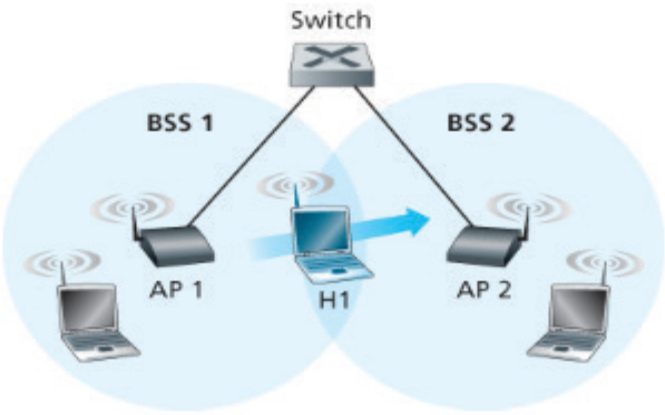


Figure 7.15 Mobility in the same subnet

Gambar 7.8 Mobilitas di subnet yang sama

5. Jaringan Area Pribadi: Bluetooth and Zigbee

› Bluetooth

Jaringan IEEE 802.15.1 beroperasi dalam jarak pendek, daya rendah, dan biaya rendah. Untuk alasan ini, jaringan 802.15.1 kadang-kadang disebut sebagai jaringan area pribadi nirkabel. Tautan dan lapisan fisik 802.15.1 didasarkan pada spesifikasi Bluetooth sebelumnya untuk personal jaringan area. Jaringan 802.15.1 beroperasi di radio tanpa izin 2,4 GHz band dengan cara TDM, dengan slot waktu 625 mikrodetik.

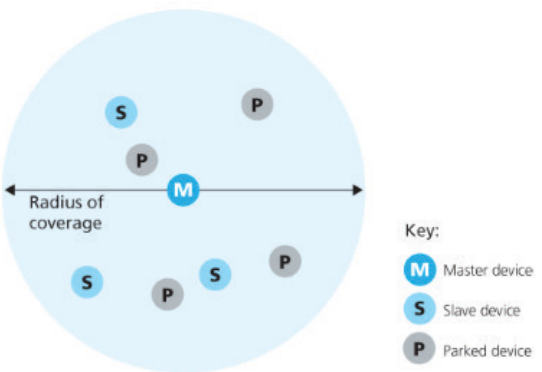


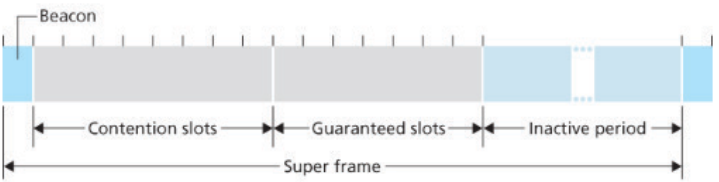
Figure 7.16 A Bluetooth piconet

Gambar 7.9 Bluetooth piconet

Selama setiap slot waktu, pengirim mentransmisikan pada salah satu dari 79 saluran, dengan saluran berubah dengan cara yang diketahui tetapi pseudo-acak dari slot ke slot. Bentuk loncatan saluran, yang dikenal sebagai spektrum sebaran frekuensi-hopping, menyebar transmisi dalam waktu melalui spektrum frekuensi.Dengan demikian, perangkat 802.15.1 harus mengatur sendiri. 802.15.1 adalah perangkat pertama kali disusun dalam piconet hingga delapan perangkat aktif.

› Zigbee

Jaringan Zigbee area pribadi kedua yang distandarisasi oleh IEEE adalah standar 802.15.4 [IEEE 802.15 2012] dikenal sebagai Zigbee. Sementara jaringan Bluetooth menyediakan kecepatan data "penggantian kabel" lebih dari satu Megabit per detik, Zigbee ditargetkan untuk aplikasi siklus kerja bertenaga lebih rendah, kecepatan data lebih rendah, dan lebih rendah daripada Bluetooth.



Gambar 7.10 Struktur superframe Zigbee 802.15.4

D. Akses Internet Seluler

1. Tinjauan Arsitektur Jaringan Seluler

Arsitektur Jaringan Seluler, 2G: Koneksi Suara ke Jaringan Telepon

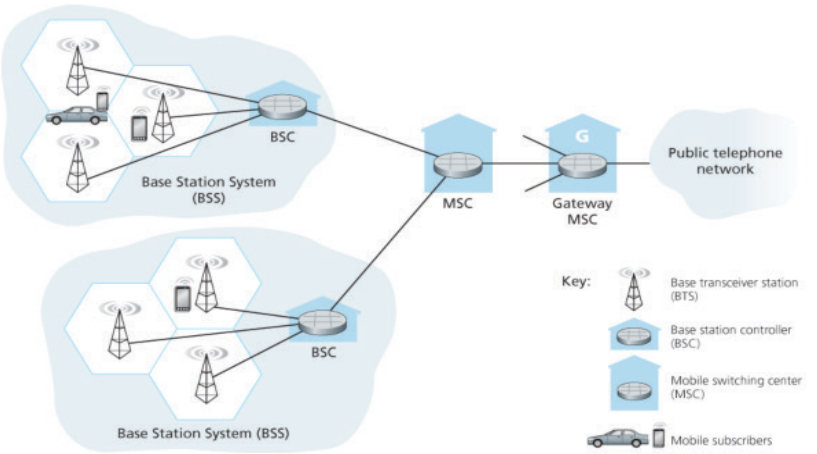


Figure 7.18 Components of the GSM 2G cellular network architecture

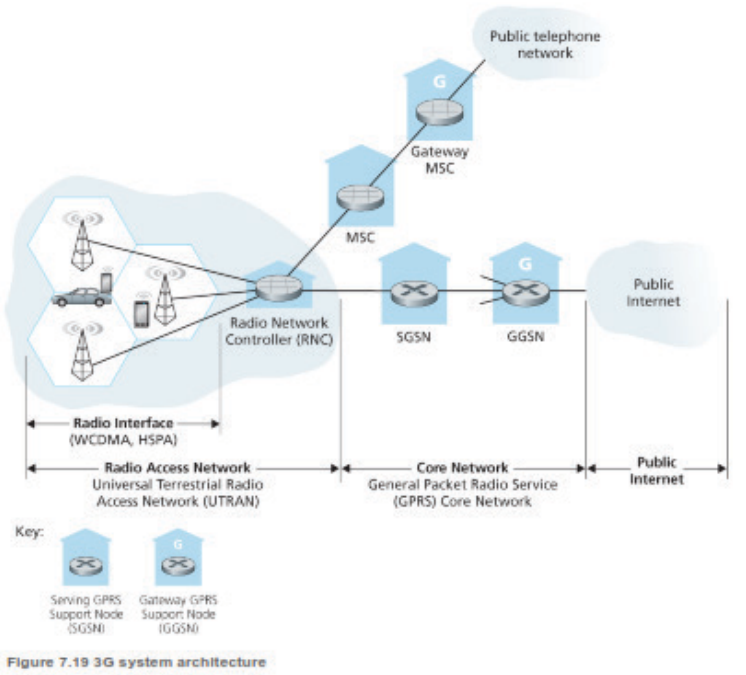
Gambar 7.11 Komponen arsitektur jaringan selular GSM 2G

Pada gambar di atas, ditunjukkan bahwa setiap sel berisi satu *best transceiver station* yang berada di tengah sel, banyak sistem saat ini menempatkan BTS di sudut-sudut dimana tiga sel berpotongan, sehingga satu BTS dengan antena pengarah dapat melayani tiga sel. Standar GSM untuk sistem seluler 2G menggunakan gabungan FDM/TDM (radio) untuk antarmuka udara. Dalam sistem FDM/TDM gabungan, saluran dipartisi menjadi beberapa sub-pita frekuensi; waktu dibagi menjadi bingkai dan slot. Sistem GSM terdiri dari pita frekuensi 200 kHz dengan masing-masing pita mendukung delapan panggilan TDM. GSM mengkodekan ucapan pada 13 kbps dan 12,2 kbps.

2. Jaringan Data Seluler 3G: Memperluas Internet ke Pelanggan Seluler

Jaringan data seluler inti 3G menghubungkan jaringan akses radio ke Internet publik. Inti jaringan bekerja sama dengan komponen-komponen jaringan suara seluler. Ada dua jenis node dalam jaringan inti 3G: Melayani GPRS Support Nodes dan Gateway Support Nodes.

SGSN bertanggung jawab untuk mengirimkan datagram ke / dari mobile node di radio akses jaringan yang dilampirkan SGSN. Internet. GGSN adalah bagian terakhir dari infrastruktur 3G yang datagram berasal dari mobile node bertemu sebelum memasuki Internet yang lebih besar.



Gambar 7.12 Arsitektur Sistem 3G

3G Radio Access Network: The Wireless Edge

Jaringan akses radio 3G adalah jaringan hop pertama nirkabel yang kita lihat sebagai pengguna 3G. *Radio Network Controller* (RNC) biasanya mengontrol beberapa stasiun transceiver basis sel yang mirip dengan basis stasiun yang kami temui dalam sistem 2G. Setiap tautan nirkabel sel beroperasi antara node seluler dan stasiun transceiver dasar, seperti di jaringan 2G. RNC terhubung ke kedua sakelar sirkuit jaringan suara seluler melalui MSC, dan ke Internet packet-switched melalui SGSN. Jadi, sementara 3G layanan suara seluler dan data seluler menggunakan jaringan inti yang berbeda, mereka berbagi hop pertama / terakhir yang sama jaringan akses radio.

Perubahan signifikan dalam 3G UMTS melalui jaringan 2G adalah daripada menggunakan FDMA / TDMA GSM skema, UMTS menggunakan teknik CDMA yang dikenal sebagai Direct Sequence Wideband CDMA. Perubahan ini membutuhkan jaringan akses nirkabel seluler 3G baru yang beroperasi secara paralel dengan jaringan radio 2G BSS. Layanan data yang terkait dengan spesifikasi WCDMA dikenal sebagai HSPA dan menjanjikan kecepatan data downlink hingga 14 Mbps. Detail tentang 3G jaringan dapat ditemukan di situs Web 3rd Generation Partnership Project.

3. On to 4G: LTE

Arsitektur Sistem: Jaringan Inti Semua-IP

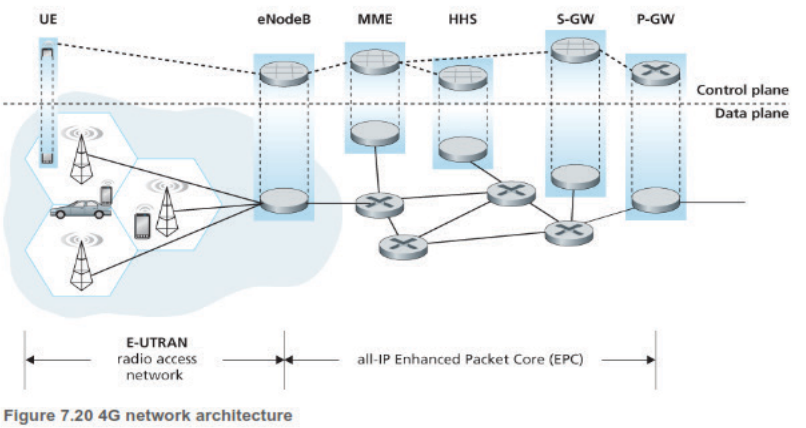


Figure 7.20 4G network architecture

Gambar 7.13 4G *Network Architecture*

Sistem seluler generasi keempat (4G) semakin banyak digunakan. Pada 2015, lebih dari 50 negara. memiliki cakupan 4G melebihi 50%. Standar 4G Jangka Panjang Evolusi (LTE) [Sauter 2014] menempatkan maju oleh 3GPP memiliki dua inovasi penting atas sistem 3G jaringan inti semua-IP dan sebuah peningkatan jaringan akses radio. komponen jaringan Ada dua level tinggi yang penting pengamatan tentang arsitektur 4G:

- Arsitektur jaringan semua-IP yang terpadu. Dengan 4G, sisa-sisa terakhir dari seluler akar jaringan di telepon telah menghilang, memberi jalan bagi layanan IP universal
- Pemisahan yang jelas antara bidang data 4G dan bidang kendali 4G. Mencerminkan perbedaan data dan pesawat kendali untuk

lapisan jaringan IP. jaringan 4G arsitektur juga dengan jelas memisahkan data dan bidang kontrol.

› Pemisahan yang jelas antara jaringan akses radio, dan semua-inti-IP jaringan. IP datagram yang membawa data pengguna diteruskan antara pengguna (UE) dan gateway melalui jaringan IP 4G-internal ke Internet eksternal.

Komponen utama arsitektur 4G adalah sebagai berikut :

› ENodeB adalah keturunan logis dari base station 2G dan Pengontrol Jaringan Radio 3G dan memainkan peran sentral. Peran data-plane adalah untuk meneruskan datagram antara UE dan P-GW. Datagram UE dienkapsulasi di eNodeB dan diteruskan ke P-GW melalui jaringan 4G all-IP enhanced packet core.

› Gateway Jaringan Data Paket (P-GW) mengalokasikan alamat IP ke UE dan melakukan QoS. Sebagai titik akhir terowongan, ia juga melakukan enkapsulasi / dekapsulasi datagram saat meneruskan datagram ke / dari UE

› Serving Gateway (S-GW) adalah titik jangkar mobilitas pesawat data — semua lalu lintas UE akan melewatinya S-GW. S-GW juga melakukan fungsi pengisian / penagihan dan intersepsi lalu lintas yang sah.

› Entitas Manajemen Mobilitas (MME) melakukan manajemen koneksi dan mobilitas atas nama dari UE di dalam sel yang dikontrolnya.

› Home Subscriber Server (HSS) berisi informasi UE termasuk akses roaming kemampuan, kualitas profil layanan, dan informasi otentikasi.

Jaringan Akses Radio LTE

LTE menggunakan kombinasi multiplexing pembagian frekuensi dan multiplexing pembagian waktu pada saluran hilir, dikenal sebagai orthogonal frequency division multiplexing.

E. IP Seluler

Arsitektur Internet dan protokol untuk mendukung mobilitas, secara kolektif dikenal sebagai IP mobile, adalah didefinisikan terutama dalam RFC 5944 untuk IPv4.Dengan demikian, IP seluler adalah standar yang kompleks.

- Standar IP seluler terdiri dari tiga bagian utama:
  - Penemuan agen. IP seluler mendefinisikan protokol yang digunakan oleh agen rumah atau asing untuk mengiklankannya layanan ke node seluler, dan protokol untuk node seluler untuk meminta layanan dari orang asing atau rumah agen.
  - Pendaftaran dengan agen rumah. Mobile IP mendefinisikan protokol yang digunakan oleh mobile node dan / atau agen asing untuk mendaftar dan membatalkan pendaftaran COA dengan agen rumah simpul seluler.
  - Perutean data tidak langsung dari datagram. Standar ini juga mendefinisikan cara di mana datagram berada diteruskan ke node seluler oleh agen rumah, termasuk aturan untuk meneruskan datagram, aturan untuk menangani kondisi kesalahan, dan beberapa bentuk enkapsulasi.
- Penemuan Agen

  Penemuan agen adalah sebuah proses penemuan agen asing baru dengan alamat jaringan baru, yang memungkinkan lapisan dalam node seluler mengetahui bahwa ia telah pindah ke jaringan asing baru. Penemuan agen dapat dilakukan dengan salah satu dari dua cara: melalui iklan agen atau melalui agen permohonan. Dengan iklan agen, agen asing atau rumah mengiklankan layanannya menggunakan ekstensi ke protokol penemuan router yang ada. Hal yang penting bidang dalam ekstensi adalah sebagai berikut:
  - Agen home bit (H). Menunjukkan bahwa agen tersebut adalah agen rumah untuk jaringan di mana ia berada.
  - Foreign agent bit (F). Menunjukkan bahwa agen tersebut adalah agen asing untuk jaringan di mana ia berada.

› Registration required bit (R). Menunjukkan bahwa pengguna seluler di jaringan ini harus mendaftar dengan a agen asing.

› M, bit enkapsulasi G. Tunjukkan apakah bentuk enkapsulasi selain IP-in-IP enkapsulasi akan digunakan.

› *Care-of address* (COA) *fields*. COA akan dikaitkan dengan agen asing, yang akan menerima datagram dikirim ke COA dan kemudian meneruskannyake node seluler yang sesuai.

F. Mengelola Mobilitas di Jaringan Seluler

Jaringan 4G pada prinsipnya mirip dengan yang digunakan dalam GSM. Seperti halnya IP seluler, bahwa sejumlah prinsip dasar yang diwujudkan dalam jaringan GSM. Seperti IP seluler, GSM mengadopsi pendekatan perutean tidak langsung. Merutekan panggilan koresponden ke jaringan rumah pengguna seluler dan ke jaringan yang dikunjungi dalam GSM terminologi, jaringan rumah pengguna ponsel disebut sebagai tanah publik rumah pengguna ponsel jaringan seluler (PLMN rumah). Jaringan rumah adalah penyedia seluler yang berlangganannya dengan pengguna seluler (misalnya penyedia yang menagih pengguna untuk layanan seluler bulanan).

Seperti halnya IP seluler, tanggung jawab jaringan rumah dan yang dikunjungi cukup berbeda.

- Jaringan rumah memelihara database yang dikenal sebagai register lokasi rumah (HLR) yang berisi nomor telepon seluler permanen dan informasi profil pelanggan untuk masing-masing pelanggan Yang penting, HLR juga berisi informasi tentang lokasi pelanggan. HLR berisi cukup informasi untuk memperoleh alamat di jaringan tempat panggilan ke pengguna seluler harus dialihkan. Saklar khusus masuk jaringan rumah, yang dikenal sebagai Gateway Mobile Services Switching Center (GMSC).
- Jaringan yang dikunjungi menyimpan basis data yang dikenal sebagai register lokasi pengunjung (VLR). VLR berisi entri untuk setiap pengguna seluler di bagian jaringan yang dilayani oleh VLR. Entri VLR datang dan pergi saat pengguna seluler masuk dan keluar dari jaringan. VLR biasanya berlokasi bersama dengan mobile switching

center (MSC) yang mengoordinasikan pengaturan panggilan dan dari jaringan yang dikunjungi.

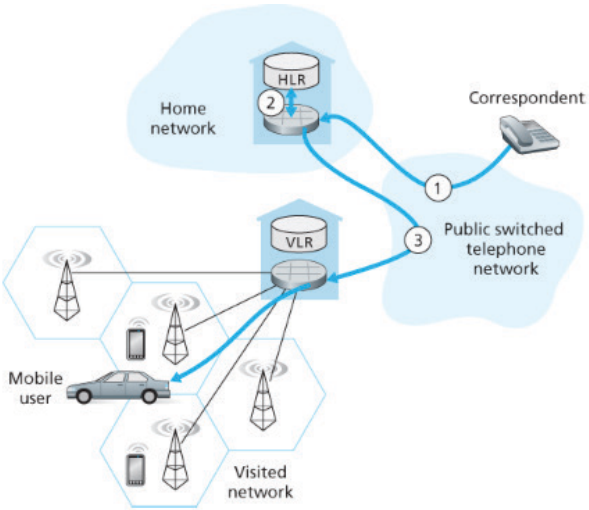
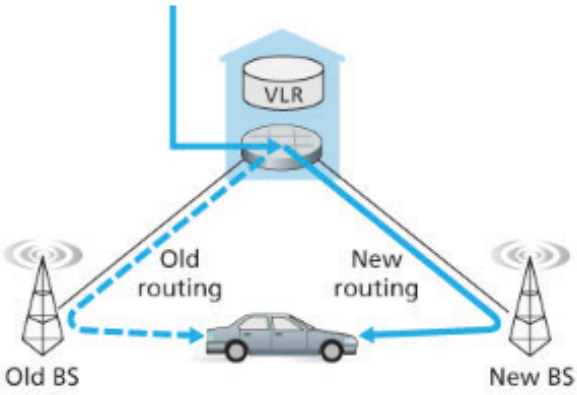


Figure 7.30 Placing a call to a mobile user: Indirect routing

Gambar 7.14 Melakukan panggilan ke pengguna seluler: Perutean tidak langsung

G. Handoffs dalam GSM

Handoff terjadi ketika stasiun bergerak mengubah hubungannya dari satu stasiun pangkalan ke stasiun pangkalan lainnya selama panggilan.



Gambar 7.15 Skenario *handoff* antara *BTS* dengan *MSC* Umum

- Kesamaan antara IP mobile dan mobilitas GSM

  Elemen GSM Mengomentari elemen. Mobile IP element Home system. Jaringan tempat nomor telepon permanen pengguna ponsel. Home network Gateway mobile switching center or simply home MSC, Home location register (HLR). Rumah MSC: titik kontak untuk mendapatkan alamat routable dari pengguna ponsel HLR: database dalam sistem rumah yang berisi nomor telepon permanen, informasi profil, lokasi pengguna ponsel, informasi berlangganan. Home agent.

- Visited mobile services switching center, Visitor location register (VLR).

  MSC yang Dikunjungi: bertanggung jawab untuk mengatur panggilan ke / dari seluler node dalam sel yang terkait dengan MSC. VLR: database sementara entri dalam sistem yang dikunjungi, berisi informasi berlangganan untuk setiap pengguna seluler yang mengunjungi. Foreign agent.

- Mobile station roaming number (MSRN) or simply roaming number.

  Alamat yang dapat dirutekan untuk segmen panggilan telepon antara rumah MSC dan mengunjungi MSC, tidak terlihat oleh ponsel maupun koresponden. Care-of address. Perbandingan manajemen mobilitas dalam GSM dan IP Seluler IP dan jaringan seluler berbeda secara fundamental dalam banyak hal, berbagi sejumlah elemen fungsional umum dan pendekatan keseluruhan dalam menangani mobilitas.

H. Nirkabel dan Mobilitas: Dampak pada Protokol Lapisan Tinggi

Jaringan nirkabel berbeda secara signifikan dari rekan-rekan kabelnya di kedua lapisan tautan dan pada lapisan jaringan.Lapisan jaringan menyediakan pengiriman upaya terbaik yang sama model layanan ke lapisan atas di jaringan kabel dan nirkabel. Protokol TCP atau UDP digunakan untuk menyediakan layanan transport-layer ke aplikasi di jaringan kabel dan nirkabel. TCP dan UDP dapat beroperasi di jaringan dengan tautan nirkabel. Di sisi lain, protokol transport masuk umum, dan TCP khususnya.

TCP mentransmisikan kembali segmen yang hilang atau rusak di jalur antara pengirim dan Penerima. Dalam kasus pengguna ponsel, kehilangan dapat terjadi akibat kemacetan jaringan (buffer router overflow) atau dari handoff. Dalam semua kasus, ACK penerima-ke-pengirim TCP hanya menunjukkan bahwa suatu segmen tidak diterima utuh. pengirim tidak mengetahui apakah segmen itu hilang karena kemacetan, selama handoff, atau karena kesalahan bit yang terdeteksi. Dalam semua kasus, respons pengirim adalah sama, untuk mengirim ulang segmen. Respons kontrol kemacetan TCP juga sama dalam semua kasus TCP mengurangi responsnya jendela kemacetan. Dengan tanpa syarat mengurangi jendela kemacetannya, TCP secara implisit mengasumsikan bahwa kehilangan segmen dihasilkan dari kemacetan daripada korupsi atau handoff.

Kesalahan bit jauh lebih umum di jaringan nirkabel daripada di jaringan kabel. Ketika kesalahan bit tersebut terjadi atau ketika kehilangan handoff terjadi, sebenarnya tidak ada alasan bagi pengirim TCP untuk melakukannya mengurangi jendela kemacetannya (dan dengan demikian menurunkan tingkat pengirimannya). Masalahnya bahwa buffer router kosong dan paket mengalir di sepanjang jalur ujung ke ujung tanpa terhalang oleh kemacetan. Peneliti menyadari pada awal hingga pertengahan 1990-an bahwa diberikan tingkat kesalahan bit tinggi pada tautan nirkabel dan kemungkinan handoff loss, respons kontrol-kemacetan TCP bisa bermasalah dalam pengaturan nirkabel. Tiga kelas pendekatan yang luas dimungkinkan untuk menangani masalah ini:

- **Pemulihan Lokal.** Dalam pendekatan pemulihan lokal, pengirim TCP adalah sangat tidak menyadari bahwa segmennya melintasi tautan nirkabel. Pendekatan alternatif adalah untuk Pengirim dan penerima TCP menyadari keberadaan tautan nirkabel, untuk membedakannya kerugian kongestif yang terjadi pada jaringan kabel dan kerugian yang terjadi pada tautan nirkabel dan untuk menerapkan kontrol kemacetan hanya sebagai respons terhadap kerugian jaringan kabel kongestif.
- **Pendekatan koneksi terpisah.** Dalam pendekatan koneksi terpisah, ujung ke ujung koneksi antara pengguna ponsel dan titik akhir lainnya dipecah menjadi dua lapisan transport koneksi: satu dari host seluler ke titik akses nirkabel, dan satu dari nirkabel jalur akses ke titik akhir komunikasi lainnya. koneksi TCP terpecah banyak, digunakan dalam jaringan data seluler, dan perbaikan signifikan memang dapat

dilakukan melalui penggunaan koneksi TCP split. Tautan nirkabel sering memiliki relatif bandwidth rendah Akibatnya, aplikasi yang beroperasi melalui tautan nirkabel, khususnya melalui sambungan nirkabel seluler, harus memperlakukan bandwidth sebagai commodity yang langka

BAB 8

KEAMANAN DI JARINGAN KOMPUTER

A. Apa itu Keamanan Jaringan?

Sistem keamanan jaringan komputer merupakan mesin yang digunakan dalam melakukan identifikasi dan melakukan pencegahan dari penggunaan yang tidak sesuai atau tidak sah pada jaringan komputer. Melalui sistem jaringan inilah dapat membantu untuk melakukan pencegahan dengan cara menghentikan pengguna yang tidak sesuai atau seringkali disebut sebagai penyusup.

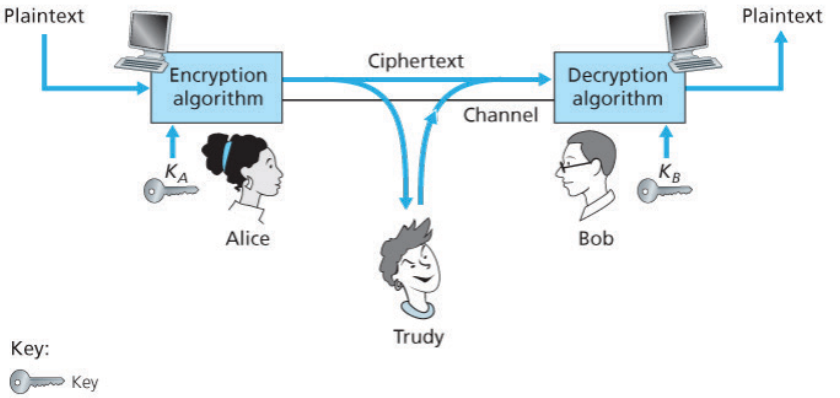
Ada hal penting yang perlu diketahui bahwa untuk jaringan tidak ada yang anti sadap maupun tidak terdapat jaringan komputer yang benar-benar aman. Hal ini dikarenakan jaringan memiliki sifat yaitu untuk melakukan komunikasi, sehingga bagi para pemiliknya harus menggunakan sistem keamanan jaringan nirkabel supaya terhindarkan dari resiko penyadapan atau hal lain yang merugikan. Untuk membentuk keamanan pada jaringan, maka ada 2 elemen yang bisa Anda ketahui, yaitu tembok pengamanan dan juga rencana pengamanan. Untuk kedua elemen tersebut akan diimplementasikan secara bersamaan dengan yang lainnya. Tujuannya yaitu supaya dapat menjaga agar sistem jaringan tidak dapat ditembus oleh pihak lain.

Hal-hal yang diinginkan dalam berkomunikasi aman:

- **Kerahasiaan.** Hanya pengirim dan penerima yang dituju yang dapat memahami kontenn pesan yang ditransmisikan. Aspek kerahasiaan

ini mungkin yang paling sering dianggap arti dari istilah komunikasi yang aman.

- **Integritas pesan.** Memastikan bahwa konten komunikasi tidak diubah, baik secara jahat atau tidak sengaja dalam perjalanan.
- **Otentikasi titik akhir.** Pengirim dan penerima harus dapat mengonfirmasi identitas pihak lain yang terlibat dalam komunikasi untuk memastikan bahwa pihak lain tersebut memang siapa atau apa mereka mengaku.
- **Keamanan operasional.** Hampir semua organisasi saat ini memiliki jaringan yang terhubung ke internet publik. Oleh karena itu, jaringan ini berpotensi menjadi dikompromikan.



Gambar 8.1 Komponen Kripto Pengirim, Penerima, dan penyusup (Alice, Bob, dan Trudy)

B. Prinsip Kriptografi

Kriptografi adalah ilmu mengenai teknik enkripsi dimana "naskah asli" (plaintext) diacak menggunakan suatu kunci enkripsi menjadi "naskah acak yang sulit dibaca" (ciphertext) oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi bisa mendapatkan kembali data asli. Probabilitas mendapat kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu yang tidak terlalu lama adalah sangat kecil. Penggunaan kriptografi untuk kerahasiaan.

1. Kriptografi Kunci Simetris

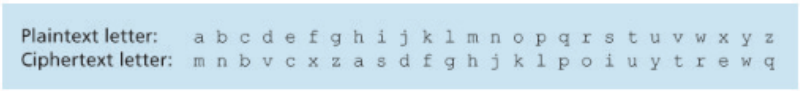


Figure 8.3 A monoalphabetic cipher

Gambar 8.2 Sandi Monoalfabet

Semua algoritma kriptografi melibatkan penggantian satu hal dengan yang lain. Algoritma kunci simetris dikenal sebagai *Caesar cipher* (sandi adalah metode untuk mengenkripsi data).

Tiga skenario membobol skema enkripsi yang bergantung pada informasi yang dimiliki penyusup:

› **Serangan hanya teks sandi**. Penyusup mungkin hanya memiliki akses ke yang dicegat ciphertext, tanpa informasi pasti tentang isi pesan teks biasa.

› **Serangan teks biasa**. Saat penyusup mengetahui beberapa (teks biasa, ciphertext).

› **Serangan teks-teks yang dipilih**. Dalam serangan teks-teks terpilih, penyusup dapat memilih pesan teks-biasa dan mendapatkan bentuk teks-teks yang sesuai.

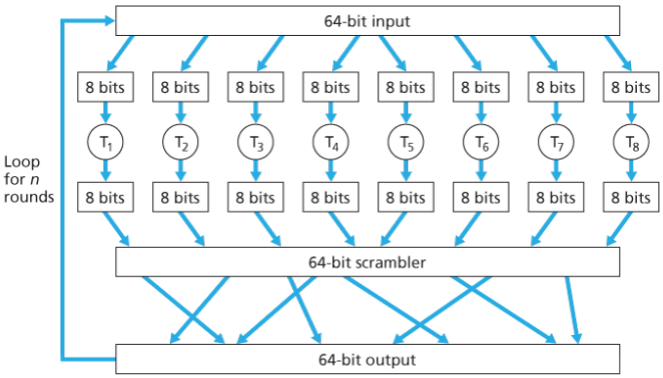
Blokir Sandi

Ada dua kelas besar dari teknik enkripsi simetris: chiper aliran dan cipher blok. Pada cipher bblok, protokol yang digunakan untuk keamanan internet termasuk PGP (untuk email aman), SSL (untuk mengamankan TCP), dan IPsec (untuk mengamankan lapisan jaringan mengangkut) cipher blok 3-bit tertentu

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Gambar 8.3 Tabel Cipher blok 3-bit tertentu

Kunci untuk algoritma ini blok akan ada delapan tabel permutasi (dengan asumsi fungsi dibagi dan dikenal secara umum).



Gambar 8.4 Contoh sandi blok

Cipher-Block Chaining

Mode operasi Cipher Block Chaining (CBC) merupakan salah satu mode operasi block cipher yang menggunakan vektor inisialisasi (initialitation vector/IV) dengan ukuran tertentu (ukurannya sama dengan satu blok plaintext). Pada mode operasi ini plaintext dibagi menjadi beberapa blok, kemudian masing-masing blok dienkripsi dengan ketentuan blok plaintext pertama dienkripsi lebih dahulu. Sebelum dienkripsi, plaintext di-XOR dengan IV. Lalu, hasil XOR tersebut dienkripsi hingga menghasilkan ciphertext. Selanjutnya, ciphertext tersebut digunakan sebagai IV untuk proses penyandian blok plaintext selanjutnya.

Mode operasi CBC menutupi kelemahan dari mode operasi ECB, karena pada CBC dapat menyembunyikan pola dari plaintext. Mengapa demikian? Karena sebelum dienkripsi, plaintext di-XOR dengan IV atau ciphertext sebelumnya, sehingga plaintext yang sama belum tentu menghasilkan ciphertext yang sama, kecuali jika memiliki IV/ciphertext sebelumnya yang sama.

CBC beroperasi sebagai berikut:

a. Sebelum mengenkripsi pesan (atau aliran data), pengirim menghasilkan k-bit acak string, yang disebut inisialisasi vektor (IV). Denote vektor inisialisasi ini oleh c (0). Pengirim mengirim IV ke penerima di cleartext.

b. Untuk blok pertama, pengirim menghitung m (1) ⊕ c (0), yaitu, menghitung eksklusif-atau dari blok pertama cleartext dengan IV. Kemudian jalankan hasilnya melalui algoritma blok-cipher untuk mendapatkan blok ciphertext yang sesuai; yaitu, c (1) = KS (m (1) ⊕ c (0)). Pengirim mengirimkan blok c (1) ke receiver.

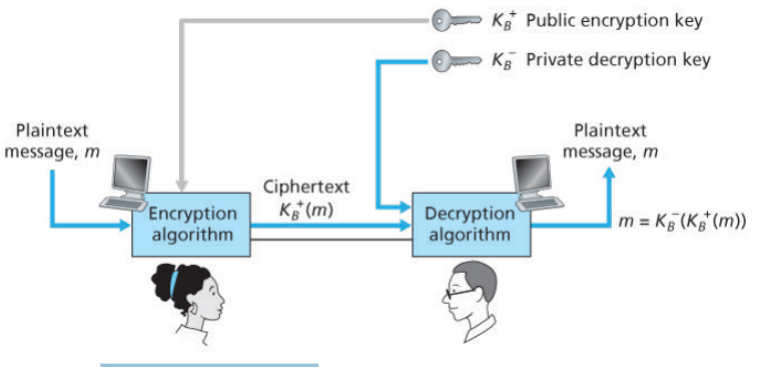
c. Untuk blok ITH, pengirim menghasilkan blok engan ciphertext dari c (i) = KS (m (i) ⊕ c (i – 1)).

2. Enkripsi Kunci Publik

Public Key Encryption adalah Sistem enkripsi (penyandian) yang menggunakan dua kunci, yaitu kunci publik dan kunci privat. Kunci publik diberitahukan oleh pemilik dan digunakan oleh semua orang yang ingin mengirimkan pesan terenkripsi kepada pemilik kunci. Kunci privat digunakan oleh pemilik kunci untuk membuka pesan terenkripsi yang ia terima.

Enkripsi kunci publik, atau kriptografi kunci publik, adalah metode mengenkripsi data dengan dua kunci berbeda dan membuat salah satu kunci, kunci publik, tersedia bagi siapa saja untuk digunakan. Kunci lainnya dikenal sebagai kunci pribadi. Data yang dienkripsi dengan kunci publik hanya dapat didekripsi dengan kunci pribadi, dan data dienkripsi dengan kunci pribadi hanya dapat didekripsi dengan kunci publik. Enkripsi kunci publik juga dikenal sebagai enkripsi asimetris. Ini banyak digunakan, terutama untuk TLS / SSL, yang memungkinkan HTTPS.

Kriptografi kunci public



Gambar 8.5 Kriptografi Kunci Publik

Plaintext Letter	*m*: numeric representation	m^e	Ciphertext c=me mod *n*
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Gambar 8.6 Tabel Enkripsi RSA Alice e=5 n=35

Table Dekripsi RSA Bob d=29 n=35

Ciphertext *c*	c^d	$m = c^d \bmod n$	Plaintext Letter
17	481968572106750915091509141182522307169 7	12	l
15	127834039403948858939111232757568359375	15	o
22	851643319086537701956194499721106030592	22	v
10	10000000000000000000000000000	5	e

Gambar 8.7 Table Dekripsi Bob's D=29, n = 35

C. Integritas Pesan dan Tanda Tangan Digital

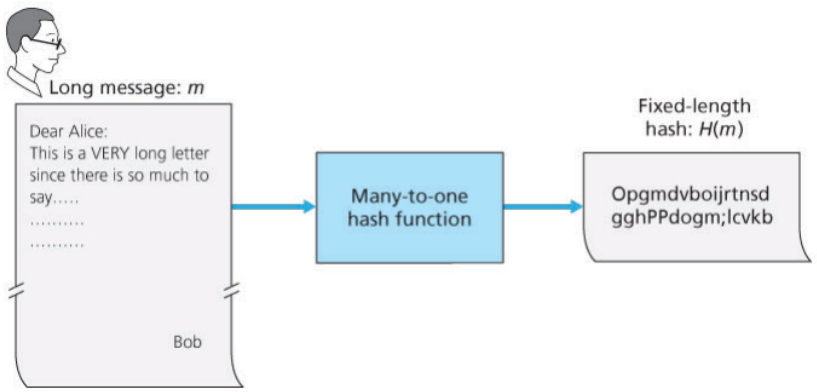
Kriptografi yang sama pentingnya untuk menyediakan integritas pesan (juga dikenal sebagai autentikasi pesan).

1. Fungsi Hash Kriptografi

Fungsi Hash Kriptografi merupakan sebuah fungsi matematis yang mengubah nilai input numerik menjadi nilai numerik yang terkompresi. Fungsi Hash sangat berguna dan muncul di hampir semua aplikasi keamanan informasi, tidak hanya di dunia kriptografi saja. Aplikasi praktis mencakup pemeriksaan integritas pesan, fingerprint digital, otentikasi, dan berbagai aplikasi keamanan informasi lainnya memakai hash function.

Informal, properti ini berarti bahwa secara komputasi tidak layak bagi penyusup untuk mengganti satu pesan untuk pesan lain yang dilindungi oleh hash.

Fungsi hash lihat dibagian gambar



Gambar 8.8 *Hash Functions*

Pesan awal dan pesan palsu memiliki checksum yang sama!

Message	ASCII Representation				
I O U 1	49	4F	55	31	
0 0 . 9	30	30	2E	39	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

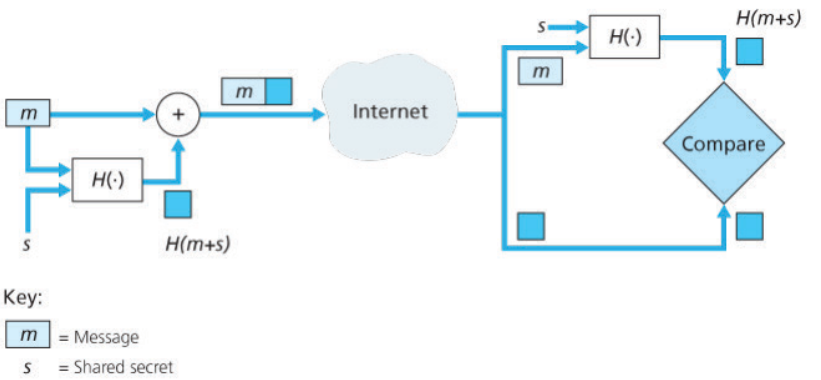
Message	ASCII Representation				
I O U 9	49	4F	55	39	
0 0 . 1	30	30	2E	31	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

Gambar 8.9 Pesan awal dan pesan penipuan memiliki checksum yang sama!

2. Kode Otentikasi Pesan

Kode otentikasi pesan adalah pesan berupa kode khusus untuk memverifikasi apakah pengguna adalah pemilik atau penyusup. Kode Otentikasi MAC- message juga dikenal sebagai hash kunci adalah cara untuk melindungi dokumen dari pemalsuan dari orang-orang yang tidak mengetahui kunci pribadi yang hanya dibagi antara pengirim

dan penerima. Motif menggunakan kode otentikasi pesan (MAC) adalah untuk mencegah kerusakan dokumen selama perjalanan. Salah satu fitur bagus dari MAC adalah tidak memerlukan algoritma enkripsi.



Gambar 8.10 Kode Otentikasi Pesan (MAC)

3. Tanda Tangan Digital

Tanda tangan Anda membuktikan fakta bahwa Anda (sebagai lawan dari orang lain) telah mengakui dan / atau menyetujui isi dokumen. Dalam dunia digital, orang sering ingin menunjukkan pemilik atau pencipta dokumen, atau untuk menandakan perjanjian seseorang dengan konten dokumen. Tanda tangan digital adalah teknik kriptografi untuk mencapai tujuan ini di dunia digital.

Tanda tangan digital adalah skema matematis yang memiliki keunikan dalam mengidentifikasi sesuatu pada umumnya dan seseorang pada khususnya. Tanda tangan digital merupakan teknik berbeda untuk memverifikasi keaslian dokumen yang dikirim secara digital. Dalam metode ini kode dilampirkan ke pesan yang juga disebut sebagai tanda tangan digital. Untuk membuat pesan, hash pesan diperlukan setelah itu dienkripsi dengan kunci pribadi pengirim. Proses ini memastikan integritas dan sumber dokumen yang dikirim. Seiring dengan kunci pribadi verifikasi hanya dapat dimungkinkan jika kunci publik yang sesuai cocok dengan sumber.

Mari kita lihat proses tanda tangan digital:

› Pesan hash dibuat oleh pengirim

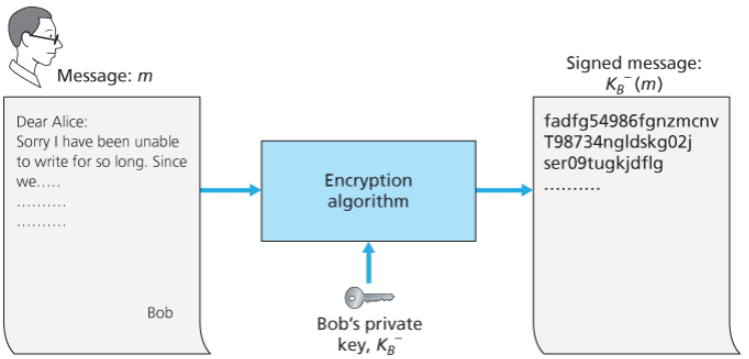
› Dia menandatangani dan meneruskan pesan hash dan (tidak terenkripsi) ke pihak yang diinginkan.

› Penerima sekarang akan menghitung hash dari pesan yang diterima dan akan mendekripsi tanda tangan.

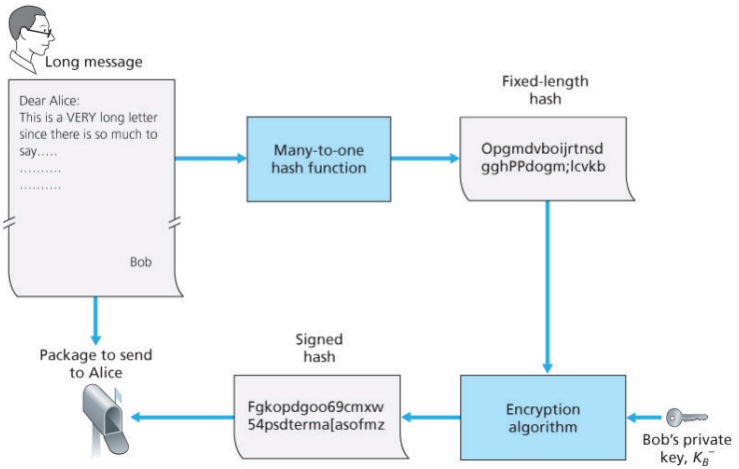
› Perbandingan dilakukan di antara pesan yang didekripsi dengan nilai hash.

› Kecocokan adalah bukti orisinalitas pesan yang dikirim dan pemberhentian segala kemungkinan perubahan.

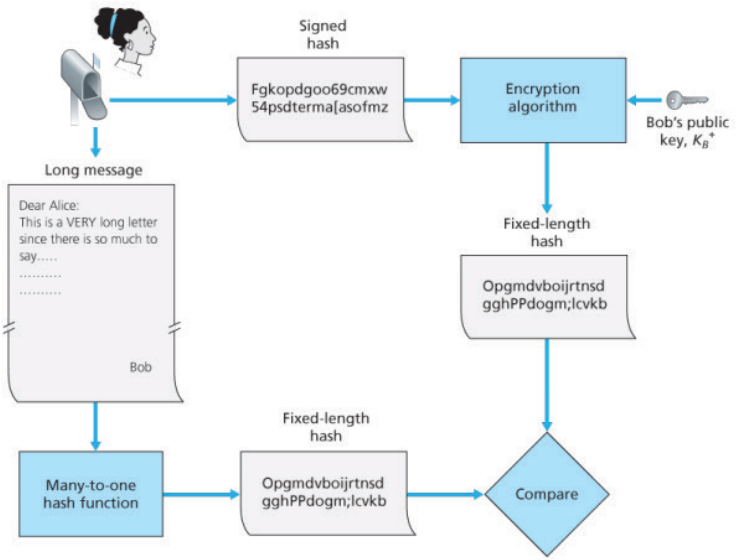
Membuat tanda tangan digital untuk dokumen



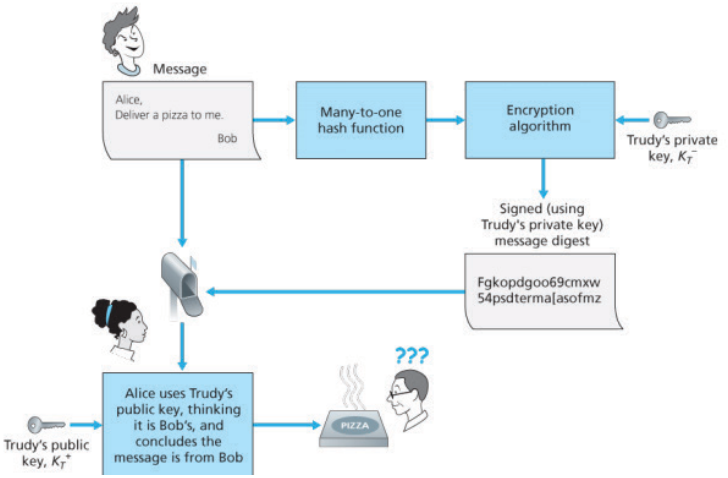
Gambar 8.11 Membuat Tanda Tangan Digital untuk Dokumen



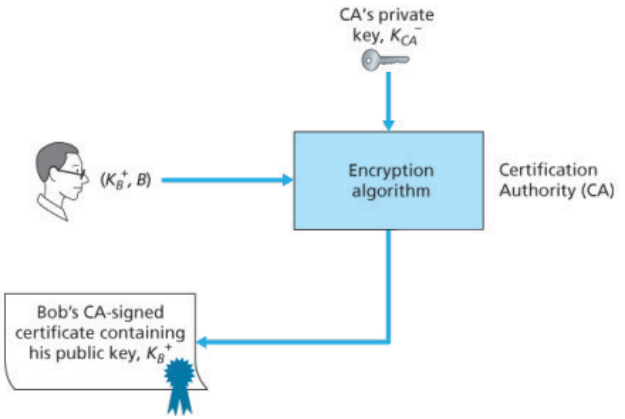
Gambar 8.12 Mengirim pesan yang ditandatangani secara digital



Gambar 8.13 Memverifikasi pesan yang ditandatangani



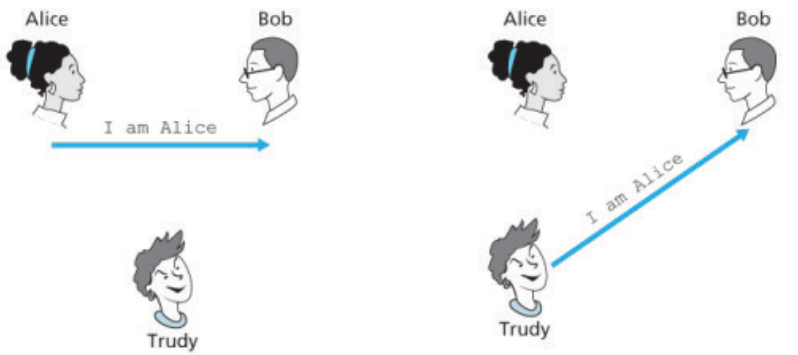
Gambar 8.14 Trudy menyamar sebagai Bob menggunakan kriptografi kunci public



Gambar 8.15 Bob memiliki kunci publik disertifikasi oleh CA

D. Otentikasi Titik Akhir

Otentikasi titik akhir adalah mekanisme keamanan yang dirancang untuk memastikan bahwa hanya perangkat resmi yang dapat terhubung ke jaringan, situs atau layanan tertentu. Pendekatan ini juga dikenal sebagai otentikasi perangkat. Dalam konteks ini, endpoint (titik akhir) yang paling sering dipertimbangkan adalah perangkat komputasi mobile, seperti laptop, ponsel pintar atau tablet tetapi bisa berupa perangkat keras yang terhubung pada jaringan TCP/IP. Segala hal yang terhubung dengan jaringan Internet termasuk komputer desktop, printer, dan perangkat keras khusus seperti server, smart meters dan smart devices lainnya.



Gambar 8.16 Protokol ap1.0 dan Skenario Kegagalan

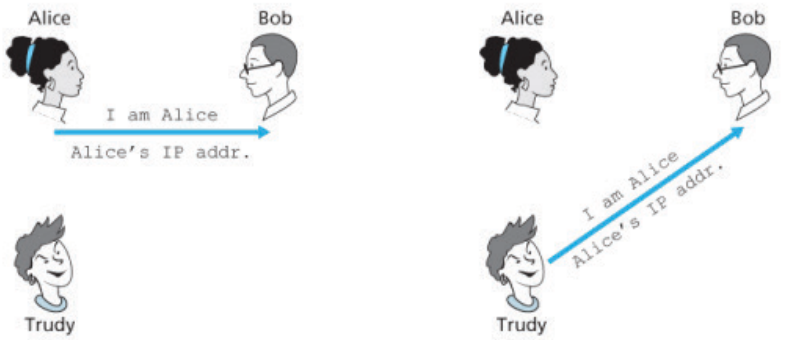
Manajemen keamanan Endpoint menjadi semakin penting dalam perluasan bidang komunikasi mesin-ke-mesin (M2M) dan *Internet of Things* (IoT). *Endpoint fingerprinting* (Sidik jari endpoint) adalah salah satu metode yang memungkinkan otentikasi titik akhir jaringan non-tradisional seperti pembaca smart card, sistem HVAC, peralatan medis, dan kunci pintu berbasis Alamat-IP. Dalam komunikasi manusia, otentikasi titik akhir sering digunakan bersama dengan otentikasi pengguna untuk keamanan yang lebih besar. Otentikasi baik pengguna dan perangkatdapat memberikan otentikasi dua faktor (2FA).

1. **Protokol Otentikasi ap1.0**

   Protokol otentikasi yang paling sederhana, karena disini hanya berisi penjelasan/pembenaran. Namun disini penyusup masih bisa masuk dan menyamar sebagai orang yang memberikan penjelasan.

2. **Protokol Otentikasi ap2.0**

   Pada protokol otentikasi ini belum ada jaringan yang dikenal, sehingga mudah sekali adanya penyusup.



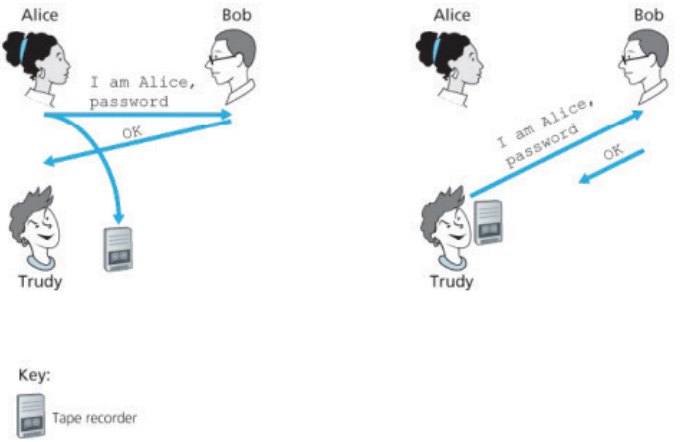
Gambar 8.17 Protokol ap2.0 dan Skenario Kegagalan

3. **Protokol Otentikasi ap3.0**

   Satu pendekatan klasik untuk otentikasi adalah dengan menggunakan kata sandi rahasia. Kata sandi adalah rahasia bersama antara autentikator dan orang yang diautentikasi. Gmail, Facebook, Telnet, FTP, dan banyak lainnya menggunakan otentikasi kata sandi.

4. Protokol Otentikasi ap3.1

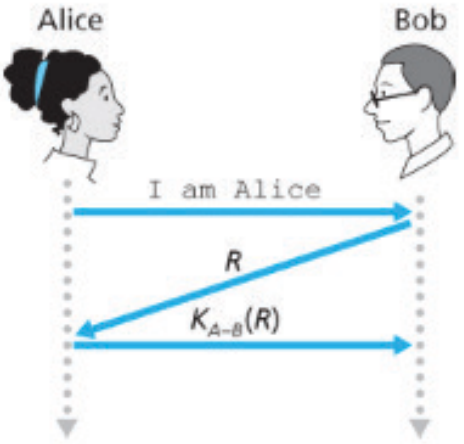
Protokol otentikasi ap3.1 adalah perbaikan dari protokol otentikasi 3.0 yaitu dengan alami untuk mengenkripsi kata sandi. Dengan mengenkripsi sandi, kita dapat mencegah Trudy dari belajar password Alice.



Gambar 8.18 Protokol ap3.0 dan Skenario Kegagalan

5. Protokol Otentikasi ap4.0

Pada protokol otentikasi ap4.0 telah ada pendeteksi klien sehingga meminimalisir adanya peyusupan. Penggunaan kriptografi kunci nonce dan simetris membentuk dasar dari ap4.0.



Gambar 8.19 Protokol ap4.0 dan Skenario Kegagalan

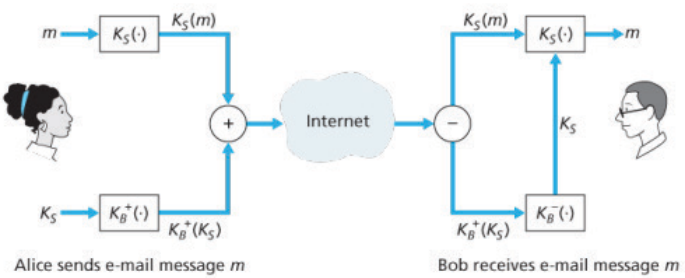
E. Mengamankan E-Mail

Ketika keamanan disediakan untuk protokol lapisan aplikasi tertentu, aplikasi yang menggunakan protokol akan menikmati satu atau lebih layanan keamanan, seperti kerahasiaan, otentikasi, atau integritas. Ketika keamanan disediakan untuk protokol transport-layer, semua aplikasi yang menggunakan protokol itu menikmati layanan keamanan dari protokol transport. Ketika keamanan disediakan pada lapisan jaringan pada basis host-ke-host, semua segmen lapisan transport (dan karenanya semua data lapisan aplikasi) menikmati layanan keamanan dari lapisan jaringan.

1. E-Mail Aman

Untuk membuat e-mail aman, digunakan prinsip-prinsip kriptografi, integritas pesan dan tanda tangan pengenal. Untuk mengatasi masalah efisiensi, mari kita gunakan sebuah kunci sesi (dibahas di bagian 8.2.2). Ttertentu, Alice (1) memilih kunci sesi simetris acak, K, (2) mengenkripsi pesannya, m, dengantombol simetris, (3) mengenkripsi kunci simetris dengan kunci publik Bob,, (4) menggabungkan pesan terenkripsi S KB dan kunci simetris terenkripsi untuk membentuk "pesan", dan (5) mengirimkan pesan ke Bob.

Alice menggunakan sesi simetris kunci, K, untuk mengirim rahasia e-mail ke Bob



Gambar 8.20 Alice Menggunakan Kunci Sesi Simetris, K, Untuk Mengirim Email Rahasia Ke Bob

2. PGP (Pretty Good Privacy)

Ketika PGP diinstal, perangkat lunak membuat pasangan kunci publik untuk pengguna. Kunci publik dapat diposting di situs Web pengguna atau ditempatkan di server kunci publik. Kunci pribadi dilindungi oleh penggunaan kata sandi. Kata sandi harus dimasukkan setiap kali

pengguna mengakses kunci pribadi. PGP memberi pengguna opsi untuk menandatangani pesan secara digital, mengenkripsi pesan, atau keduanya menandatangani dan mengenkripsi secara digital. Pesan ini muncul setelah tajuk MIME. Data yang dikodekan dalam pesan adalah, yaitu intisari pesan yang ditandatangani secara digital. PGP adalah contoh e-mail yang bagus skema enkripsinya.

```
-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.0
u2R4d+/jKmn8Bc5+hgDsqAewsDfrGdszX68liKm5F6Gc4sDfcXyt
RfdS10juHgbcfDssWe7/K=lKhnMikLo0+1/BvcX4t==Ujk9PbcD4
Thdf2awQfgHbnmKlok8iy6gThlp
-----END PGP MESSAGE
```

Gambar 8.21 Pesan Masuk PGP

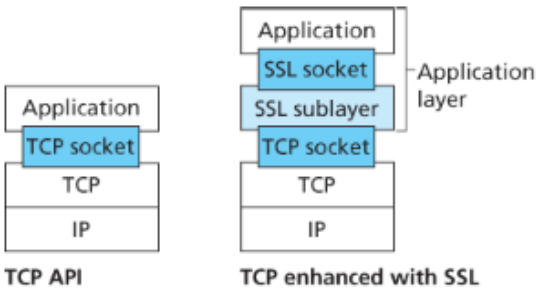
```
-----BEGIN PGP SIGNED MESSAGE-----
Hash:  SHA1
Bob:
Can I see you tonight?
Passionately yours, Alice
-----BEGIN PGP SIGNATURE-----
Version: PGP for Personal Privacy 5.0
Charset:  noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJhFEvZP9t6n7G6m5Gw2
-----END PGP SIGNATURE-----
```

Gambar 8.22 Pesan Rahasia PGP

F. Mengamankan Koneksi TCP: SSL

1. Gambaran Besar

Gambaran besar tentang koneksi TCP adalah dengan menyederhanakan SSL sebagai 'almost SSL'.



Gambar 8.23 Meskipun Ssl Secara Teknis Berada di Lapisan Aplikasi, dari Perspektif Pengembang Itu Adalah Protokol Lapisan Transport

Handshake

Membangun koneksi TCP untuk memastikan bahwa pengguna adalah user asli.

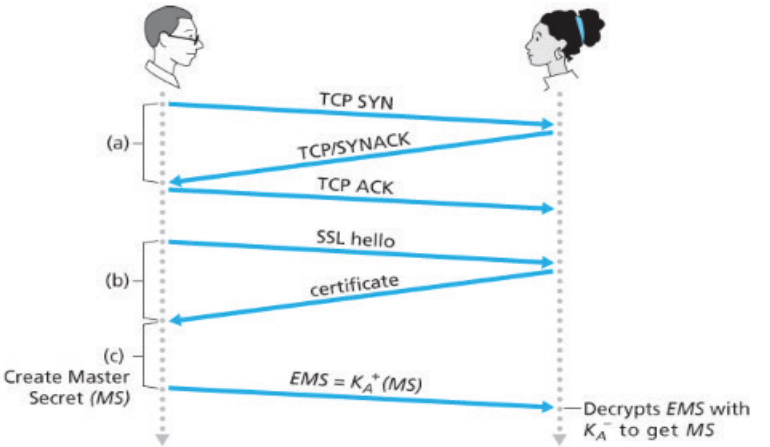


Figure 8.25 The almost-SSL handshake, beginning with a TCP connection

Gambar 8.24 Jabat tangan yang hampir SSL, dimulai dengan koneksi TCP

Asal Kunci

Transfer Data

Berbagi kunci yang sama sehingga dapat mengirim mengamankan data satu sama lain melalui koneksi TCP.

Rekaman SSL

Catatan SSL yang terdiri dari jenis bidang, bidang versi, bidang panjang, bidang data, dan bidang MAC.

2. Gambaran yang Lebih Lengkap

Gambar 8.25 Format rekaman untuk SSL

G. Keamanan Lapisan Jaringan: Ipsec dan Virtual Private Network

Protokol keamanan IP, lebih dikenal sebagai IPsec, menyediakan keamanan di lapisan jaringan. IPsec mengamankan datagram IP antara dua entitas lapisan jaringan, termasuk host dan router. Seperti yang akan segera kami jelaskan, banyak lembaga (perusahaan, cabang pemerintah, organisasi nirlaba, dan sebagainya) menggunakan IPsec untuk membuat jaringan pribadi virtual (VPN) yang dijalankan melalui Internet publik.

BAB 9
JARINGAN MULTIMEDIA

Orang-orang di seluruh penjuru dunia saat ini menggunakan Internet untuk menonton film dan acara televisi sesuai permintaan. Film Internet dan distribusi televisi perusahaan seperti Netflix dan Amazon di Amerika Utara dan Youku dan Kankan di Cina secara praktis menjadi nama rumah tangga. Tetapi orang-orang tidak hanya menonton video Internet, mereka juga menggunakan YouTube seperti untuk mengunggah dan mendistribusikan konten buatan pengguna mereka sendiri, menjadi videoproduser Internet serta konsumen. Selain itu, aplikasi jaringan seperti Skype, Google Talk, dan WeChat (sangat populer di Cina) memungkinkan orang untuk tidak hanya membuat "panggilan telepon" melalui Internet, tetapi juga untuk meningkatkan panggilan dengan konferensi video dan multi-orang.

A. Aplikasi Jaringan Multimedia

1. Properti Video

Karakteristik video yang paling menonjol adalah bit rate-nya yang tinggi. Video yang didistribusikan melalui Internettypically berkisar dari 100 kbps untuk konferensi video berkualitas rendah hingga lebih dari 3 Mbps untuk streaming film definisi tinggi. Untuk memahami bagaimana tuntutan bandwidth video dibandingkan dengan aplikasi Internet lainnya, mari kita pertimbangkan secara singkat tiga pengguna yang berbeda, masing-masing menggunakan aplikasi Internet yang

berbeda. Pengguna pertama kita, Frank, akan dengan cepat melalui foto yang diposting di halaman Facebook teman-temannya. Mari kita asumsikan bahwa Frank melihat foto baru setiap 10 detik, dan bahwa foto berukuran rata-rata 200 Kbytes. (Seperti biasa, sepanjang diskusi ini kita membuat asumsi penyederhanaan itu 1 Kbyte = 8.000 bit.)

Pengguna kedua kita, Martha, streaming musik dari Internet ("cloud") ke smartphone-nya. Mari kita anggap Martha menggunakan layanan seperti Spotify untuk mendengarkan banyak lagu MP3, satu demi satu, masing-masing dikodekan pada kecepatan 128 kbps. Pengguna ketiga kita, Victor, sedang menonton video yang telah dikodekan pada 2 Mbps. Akhirnya, misalkan panjang sesi untuk ketiga pengguna adalah 4.000 detik (sekitar 67 menit). **Tabel 9.1** membandingkan laju bit dan total byte yang ditransfer untuk ketiga pengguna ini. Jika streaming video menghabiskan bandwidth paling banyak, memiliki bit rate lebih dari sepuluh kali lebih besar daripada aplikasi streaming musik dan Facebook.

Tabel 9.1 Perbandingan Persyaratan Laju Bit dari Tiga Aplikasi Internet

	kecepatan bit	Bit ditransfer dalam 67 menit
Facebook Frank	160 kbps	80 mbit
Martha Music	128 kbps	64 mbit
Victor Vidio	2 mbps	1 gbit

Dalam aplikasi video jaringan, hal pertama yang harus kita ingat adalah persyaratan bit-rate tinggi dari video. Mengingat popularitas video dan bit rate-nya yang tinggi, mungkin tidak mengherankan jika Cisco memprediksi [Cisco 2015] bahwa streaming dan penyimpanan video akan menjadi sekitar 80 persen dari lalu lintas Internet konsumen global pada tahun 2019.

Karakteristik penting lain dari video adalah dapat dikompresi, sehingga memperdagangkan kualitas video dengan bit rate. Video adalah urutan gambar, biasanya ditampilkan pada kecepatan konstan, misalnya, pada 24 atau 30 gambar per detik. Gambar yang tidak terkompresi dan dikodekan secara digital terdiri dari array piksel,

dengan setiap piksel dikodekan ke dalam sejumlah bit untuk mewakili pencahayaan dan warna.

2. Properti Audio

Audio digital (termasuk suara dan musik digital) memiliki kebutuhan bandwidth yang jauh lebih rendah daripada video. Untuk memahami properti ini, pertama-tama pertimbangkan bagaimana audio analog (yang dihasilkan manusia dan instrumen musik) dikonversi menjadi sinyal digital :

- Sinyal audio analog disampel pada tingkat tertentu, misalnya, pada 8.000 sampel per detik. Nilai setiap sampel akan berupa bilangan real.
- Masing-masing sampel kemudian dibulatkan ke salah satu dari sejumlah nilai yang terbatas. Operasi ini disebut sebagai kuantisasi. Jumlah nilai terbatas seperti itu disebut nilai kuantisasi biasanya merupakan kekuatan dua, misalnya, 256 nilai kuantisasi
- Setiap nilai kuantisasi diwakili oleh jumlah bit yang tetap. Misalnya, jika ada256 nilai kuantisasi, maka setiap nilai dan karenanya setiap sampel audio diwakili oleh onebyte. Representasi bit dari semua sampel kemudian digabungkan bersama untuk membentuk representasi digital dari sinyal.

Audio compact disk (CD) juga menggunakan PCM, dengan laju pengambilan sampel sebesar 44.100 sampel per detik dengan 16 bit per sampel; ini memberikan tingkat 705,6 kbps untuk mono dan 1,411Mbps untuk stereo.

Teknik kompresi digunakan untuk mengurangi laju bit aliran. Kemampuan berbicara manusia dapat ditekan hingga kurang dari 10 kbps dan masih dapat dipahami. Teknik kompresi populer untuk musik stereo berkualitas dekat CD adalah MPEG 1 layer 3, lebih dikenal sebagai MP3. Encoders MP3 dapat dikompres ke banyak tingkat yang berbeda; 128 kbps adalah laju penyandian yang paling umum dan menghasilkan degradasi suara yang sangat sedikit.

Meskipun laju bit audio umumnya jauh lebih rendah daripada kecepatan video, pengguna umumnya lebih sensitif terhadap gangguan audio daripada gangguan video.

3. Jenis Aplikasi Jaringan Multimedia

Internet mendukung beragam aplikasi multimedia yang bermanfaat dan menghibur. Dalam bagian ini, kita mengklasifikasikan aplikasi multimedia ke dalam tiga kategori besar:

- streaming audio/video tersimpan
- percakapan suara/video-over-IP
- streaming audio/video langsung

a. Streaming Audio dan Video Tersimpan

Dalam kelas aplikasi ini, media yang mendasarinya adalah video yang direkam sebelumnya, seperti film, acara televisi, acara olahraga yang direkam sebelumnya, atau video yang dibuat pengguna yang direkam sebelumnya (seperti yang biasa dilihat di YouTube). Video yang direkam sebelumnya ditempatkan di server, dan pengguna mengirim permintaan ke server untuk melihat video sesuai permintaan. Banyak perusahaan internet saat ini menyediakan video streaming, termasuk YouTube (Google), Netflix, Amazon, dan Hulu.

b. Suara Percakapan dan Video-over-IP

Suara percakapan waktu-nyata melalui Internet sering disebut sebagai telepon Internet, karena, dari sudut pandang pengguna, ini mirip dengan layanan telepon tradisional. Ini juga biasa disebut Voice-over-IP (VoIP). Video percakapan serupa, kecuali bahwa itu termasuk video para peserta serta suara mereka. Sebagian besar sistem percakapan suara dan video saat ini memungkinkan pengguna untuk membuat konferensi dengan tiga atau lebih peserta. Suara dan video percakapan banyak digunakan di Internet saat ini, dengan perusahaan-perusahaan Internet Skype, QQ, dan Google Talk.

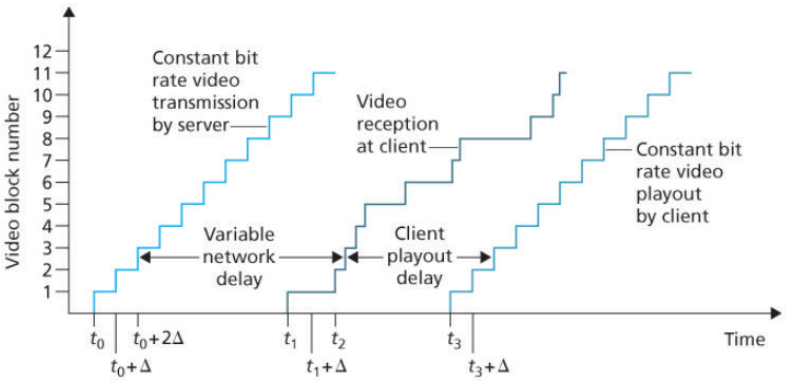
c. Streaming Audio dan Video Langsung

Aplikasi kelas ketiga ini mirip dengan siaran radio tradisional dan televisi, kecuali pengiriman yang dilakukan melalui Internet. Aplikasi-aplikasi ini memungkinkan pengguna untuk menerima transmisi radio atau televisi secara langsung seperti acara olahraga langsung atau acara berita yang sedang berlangsung ditransmisikan dari pemain internet manapun di dunia.

B. Streaming Video Yang Disimpan

Untuk streaming aplikasi video, video yang direkam sebelumnya ditempatkan di server, dan pengguna mengirim permintaan ke server untuk melihat video sesuai permintaan. Pengguna dapat menonton video dari awal hingga akhir tanpa gangguan, memungkinkan berhenti menonton video jauh sebelum vidio berakhir, atau berinteraksi dengan video mengubah posisi ke awal atau akhir. Sistem streaming video dapat diklasifikasikan ke dalam tiga kategori: Streaming UDP, streaming HTTP, dan streaming HTTP adaptif.

Karakteristik umum dari ketiga bentuk streaming video adalah penggunaan buffering aplikasi sisi klien yang luas untuk mengurangi efek dari berbagai penundaan end-to-end dan berbagai jumlah bandwidth yang tersedia antara server dan klien. Untuk streaming video (baik disimpan dan hidup), pengguna biasanya dapat mentolerir penundaan awal beberapa detik kecil antara ketika klien meminta video dan ketika pemutaran video dimulai pada klien.



Gambar 9.1 Keterlambatan Pemutaran Klien dalam *Streaming* Video

1. Streaming UDP

Dengan streaming UDP, server mentransmisikan video pada tingkat yang sesuai dengan tingkat konsumsi video klien dengan mencatat potongan video di atas UDP pada tingkat yang stabil. Misalnya, jika tingkat konsumsi video adalah 2 Mbps dan setiap paket UDP membawa 8.000 bit video, maka server akan mengirimkan satu paket UDP ke dalam soketnya setiap (8000 bit) / (2 Mbps) = 4 Mbps

2. Streaming HTTP

Dalam streaming HTTP, video hanya disimpan di server HTTP sebagai file biasa dengan URL tertentu. Ketika pengguna ingin melihat video, klien membuat koneksi TCP dengan server dan mengeluarkan permintaan GET HTTP untuk URL tersebut. Server kemudian mengirim file video, dalam pesan tanggapan HTTP, secepat mungkin, yaitu, secepat kontrol kemacetan TCP dan kontrol aliran akan memungkinkan. Di sisi klien, bit dikumpulkan dalam buffer aplikasi klien. Setelah jumlah bit dalam buffer ini melebihi ambang yang telah ditentukan, aplikasi klien memulai pemutaran khususnya, secara berkala mengambil frame video dari buffer aplikasi klien, mendekompresi frame, dan menampilkannya di layar pengguna.

Penggunaan HTTP melalui TCP juga memungkinkan video untuk melintasi firewall dan NAT lebih mudah (yang sering dikonfigurasi untuk memblokir sebagian besar lalu lintas UDP tetapi memungkinkan sebagian besar lalu lintas HTTP). Streaming melalui HTTP juga menunjukkan perlunya server kontrol media, seperti server RTSP, mengurangi biaya penyebaran skala besar melalui Internet. Karena semua kelebihan ini, sebagian besar aplikasi streaming video termasuk YouTube dan Netflix menggunakan streaming HTTP (melalui TCP) sebagai protokol streaming yang mendasarinya.

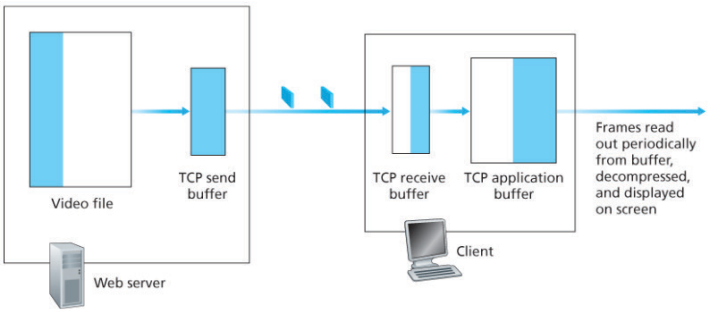
- **Pra-pengambilan video**

  Seperti yang baru saja kita pelajari, buffer sisi-klien dapat digunakan untuk mengurangi efek dari berbagai penundaan end-to-end dan memvariasikan bandwidth yang tersedia. Untuk streaming video yang disimpan, klien dapat mencoba mengunduh video pada tingkat yang lebih tinggi dari tingkat konsumsi, sehingga mengambil frame video yang akan dikonsumsi di masa depan. Pengambilan vidio ini secara alami disimpan dalam buffer aplikasi klien. Pengambilan semacam itu terjadi secara alami dengan streaming TCP, karena mekanisme penghindaran kemacetan TCP akan berusaha menggunakan semua bandwidth yang tersedia antara server dan klien.

- **Penyangga Aplikasi Klien dan Penyangga TCP**

  Di sisi server, bagian dari file video berwarna putih telah dikirim ke soket server, sedangkan bagian darkened adalah yang masih harus dikirim. Setelah "melewati pintu soket," byte ditempatkan di buffer

kirim CCP sebelum dikirim ke Internet, karena buffer pengirim TCP di sisi server ditunjukkan penuh, Server sementara dicegah dari mengirimkan lebih banyak byte dari file video ke soket. Di sisi klien, aplikasi klien (media player) membaca byte dari buffer penerimaan TCP (melalui soket kliennya) dan menempatkan byte ke buffer aplikasi klien. Pada saat yang sama, aplikasi klien secara berkala mengambil frame video dari buffer aplikasi klien, mendekompresi frame, dan menampilkannya di layar pengguna. Perhatikan bahwa jika buffer aplikasi klien lebih besar dari file video, maka seluruh proses pemindahan bit dari penyimpanan server ke buffer aplikasi klien sama dengan unduhan file biasa melalui HTTP — klien hanya menarik video dari server secepat yang diizinkan TCP!

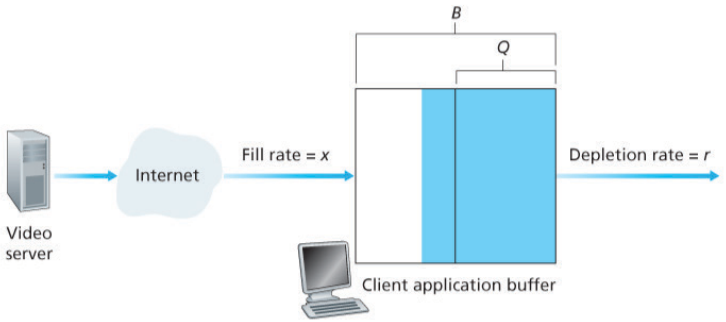


Gambar 9.2 Streaming Video yang Tersimpan melalui HTTP / TCP

Pertimbangkan sekarang apa yang terjadi ketika pengguna menjeda video selama proses streaming. Selama periode jeda, bit tidak dihapus dari buffer aplikasi klien, meskipun bit terus menghibur buffer dari server. Jika buffer aplikasi klien terbatas, akhirnya mungkin menjadi penuh, yang akan menyebabkan "tekanan balik" sepanjang perjalanan kembali ke server. Secara khusus, setelah bufferbecomes aplikasi klien penuh, bit tidak lagi dapat dihapus dari buffer menerima TCP klien, sehingga juga menjadi penuh. Setelah klien menerima buffer TCP menjadi penuh, bit tidak lagi dapat dihapus dari buffer server penyangga TCP, jadi itu juga menjadi penuh. Setelah TCP menjadi penuh, server tidak dapat mengirim bit lagi ke soket.

- **Analisis Streaming Video**

  Beberapa pemodelan sederhana akan memberikan lebih banyak wawasan tentang penundaan pemutaran awal dan pembekuan karena penipisan aplikasi buffer.



Gambar 9.3 Analisis Penyangga Sisi Klien untuk Streaming Video

- **Analisis buffer sisi klien untuk streaming video**

  Pengakhiran Awal dan Pemosisian Ulang Video

  (dalam bit) dari buffer aplikasi klien, dan biarkan Q menunjukkan jumlah bit yang harus buffered sebelum aplikasi klien mulai bermain. (tentu saja Q<B) membiarkan *r* menunjukkan tingkat konsumsi video tingkat di mana klien mengambil bit dari buffer aplikasi klien selama pemutaran. Jadi, misalnya, jika frame rate video adalah 30 frame / detik, dan setiap frame (terkompresi) adalah 100.000 bit, maka r=3 Mbps.

C. Voice-over-IP

Suara percakapan waktu-nyata melalui Internet sering disebut sebagai telepon Internet, karena, dari sudut pandang pengguna, ini mirip dengan layanan telepon tradisional. Ini juga biasa disebut Voice-over-IP (VoIP).

1. Keterbatasan Layanan IP Upaya Terbaik

Protokol lapisan jaringan Internet, IP, menyediakan layanan yang membuat upaya terbaik untuk memindahkan setiap datagram dari sumber ke tujuan secepat mungkin tetapi membuat nopromis apa pun tentang mendapatkan paket ke tujuan dalam beberapa waktu tunda atau tentang jumlah paket yang hilang.

› Paket Hilang

Pertimbangkan salah satu segmen UDP yang dihasilkan oleh aplikasi VoIP. Segmen UDP diringkas dalam datagram IP. Saat

datagram mengembara melalui jaringan, ia melewati buffer yang melalui komputer (yaitu, antrian) sambil menunggu transmisi pada tautan keluar. Ada kemungkinan bahwa satu atau lebih buffer di jalur dari pengirim ke penerima penuh, dalam hal ini datagram IP yang tiba mungkin dibuang, tidak pernah sampai pada aplikasi penerima.

Kehilangan bisa dihilangkan dengan mengirim paket melalui TCP (yang menyediakan transfer data yang andal) daripada melalui UDP. Namun, mekanisme pengiriman ulang sering dianggap tidak dapat diterima untuk aplikasi audio real-time konversi seperti VoIP, karena mereka meningkatkan penundaan ujung ke ujung

› Keterlambatan ujung ke ujung

Adalah akumulasi dari transmisi, pemrosesan, dan keterlambatan antrian dalam router; keterlambatan propagasi dalam tautan; dan keterlambatan pemrosesan sistem akhir. Untuk aplikasi percakapan waktu-nyata, seperti VoIP, keterlambatan end-to-end yang lebih kecil dari 150 msec tidak dirasakan oleh pendengar manusia; keterlambatan antara 150 dan 400 msec dapat diterima tetapi tidak ideal; dan keterlambatan melebihi 400 msecs dapat secara serius menghambat interaktivitas dalam percakapan suara.

› Paket Jitter

Sebagai contoh, pertimbangkan dua paket berturut-turut dalam aplikasi VoIP. Pengirim mengirim paket kedua 20 msecs setelah mengirim paket pertama. Tetapi pada penerima, jarak antara paket-paket ini dapat menjadi lebih besar dari 20 msecs.

2. Menghapus Jitter pada Penerima untuk Audio

Untuk aplikasi VoIP, di mana paket-paket dihasilkan secara berkala, penerima harus berusaha untuk menyediakan playout potongan suara berkala di hadapan jitter jaringan acak. Ini biasanya dilakukan dengan menggabungkan dua mekanisme berikut:

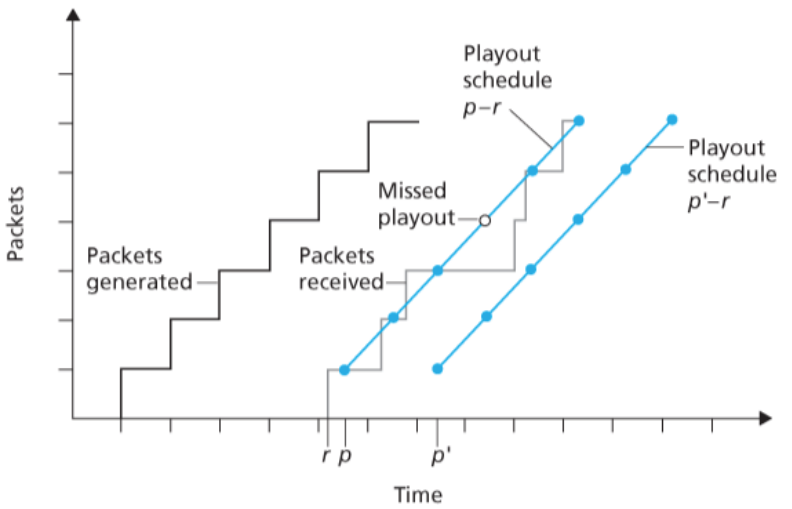
› **Siapkan setiap chunk dengan cap waktu.** Pengirim mencap setiap chunk dengan waktu di mana chunk dihasilkan.

› **Menunda pemutaran potongan pada penerima.** Penundaan pemutaran potongan audio yang diterima harus cukup lama

sehingga sebagian besar paket diterima sebelum waktu bermain yang dijadwalkan.

Memperbaiki Keterlambatan Playout

Dengan strategi penundaan tetap, penerima mencoba untuk memainkan setiap chunk persis q msecs setelah chech dihasilkan. Jadi, jika sepotong cap waktu di pengirim pada waktu t, penerima memutar keluar pada waktu t+q dengan asumsi potongan telah tiba pada saat itu. Paket yang tiba setelah waktu pemutaran yang dijadwalkan dibuang dan dianggap hilang.



Gambar 9.4 Paket Hilang untuk Penundaan Main yang Tetap Berbeda

Seperti yang ditunjukkan oleh tangga paling kiri, pengirim menghasilkan paket secara berkala, katakanlah, setiap 20 msec. Paket pertama dalam dorongan pembicaraan ini diterima pada waktu r. Seperti yang ditunjukkan pada gambar, kedatangan paket-paket berikutnya secara mencolok diberi jarak karena jitter jaringan. Mengikuti [Ramjee 1994], kita sekarang menjelaskan algoritma umum yang dapat digunakan penerima untuk menyesuaikan penundaan playout secara adaptif. Untuk tujuan ini, biarkan

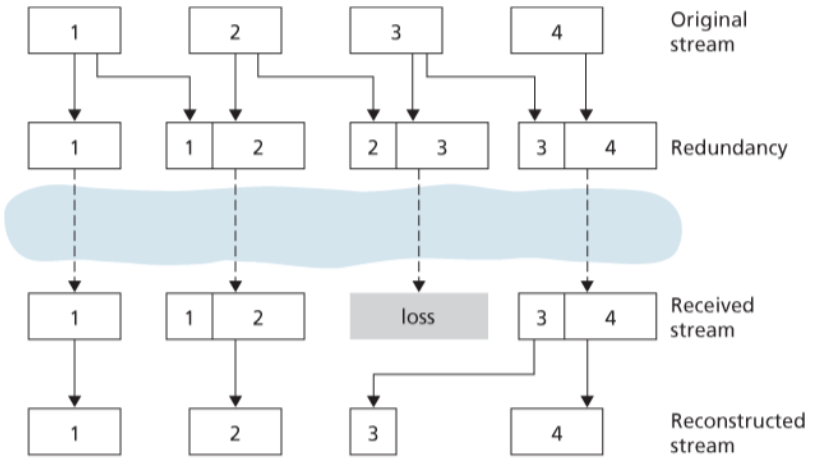
3. Memulihkan dari Kehilangan Paket

Untuk menjaga kualitas audio yang dapat diterima di hadapan packetloss. Skema semacam itu disebut skema pemulihan kerugian.

Di sini kita mendefinisikan kehilangan paket dalam arti luas: Paket hilang baik jika tidak pernah tiba di penerima atau jika tiba setelah waktu bermain yang dijadwalkan.

Forward Error Correction (FEC) / Teruskan Koreksi Kesalahan

Ide dasar FEC adalah menambahkan informasi yang berlebihan ke aliran paket asli. Untuk biaya yang secara umum meningkatkan laju transmisi, informasi yang berlebihan dapat digunakan untuk merekonstruksi pendekatan atau versi tepat dari beberapa paket yang hilang. Potongan yang redundan diperoleh dengan eksklusif ATAU potongan asli [Shacham 1990]. Dengan cara ini jika ada satu paket dari grup n+1 paket hilang, penerima dapat merekonstruksi paket yang hilang. Tetapi jika dua atau lebih paket dalam suatu kelompok hilang, penerima tidak dapat merekonstruksi paket yang hilang. Dengan menjaga n+1 ukuran kelompok, kecil, sebagian besar dari paket yang hilang dapat dipulihkan ketika kehilangan tidak berlebihan. Namun, semakin kecil ukuran grup, semakin besar peningkatan laju transmisi. Secara khusus, laju transmisi meningkat sebesar faktor 1 / n, sehingga, jika n=3 maka tingkat transmisi meningkat sebesar 33 persen.



Gambar 9.5 *Piggybacking Lower-Quality Redundant Information*

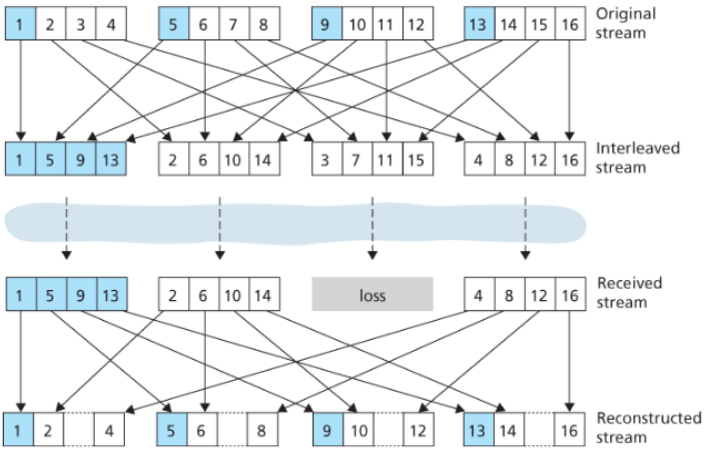
› Interleaving

Sebagai alternatif untuk transmisi redundan, aplikasi VoIP dapat mengirim audio yang disisipkan. Interleaving dapat mengurangi efek dari kehilangan paket. Jika, misalnya, panjang unit adalah 5

msecs dan chunk adalah 20 msecs (yaitu, empat unitperpotong), maka chunk pertama dapat berisi unit 1, 5, 9, dan 13; potongan kedua dapat berisi units2, 6, 10, dan 14; dan seterusnya. Gambar 9.6 menunjukkan bahwa hilangnya satu paket tunggal dari aliran interleaved menghasilkan beberapa celah kecil dalam aliran yang direkonstruksi, sebagai lawan dari celah besar tunggal yang akan terjadi dalam aliran tanpa interleaved.

› Error Concealment / Penyembunyian Kesalahan

Skema penyembunyian kesalahan berusaha menghasilkan pengganti untuk paket yang hilang yang mirip dengan aslinya. Sebagaimana dibahas dalam [Perkins 1998], ini dimungkinkan karena audio sinyal, dan dalam pidato tertentu, menunjukkan sejumlah besar kemiripan jangka pendek. Dengan demikian, teknik ini bekerja untuk tingkat kerugian yang relatif kecil (kurang dari 15 persen), dan untuk paket kecil (4-40 msecs).



Gambar 9.6 Mengirim Audio yang disisipkan

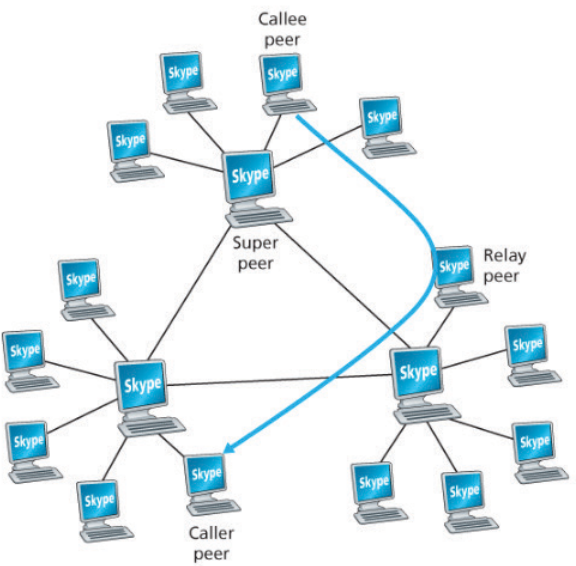
4. Studi Kasus: VoIP dengan Skype

Skype adalah aplikasi VoIP yang sangat populer dengan lebih dari 50 juta akun aktif setiap hari. Selain menyediakan layanan VoIP host-to-host, Skype menawarkan layanan host-ke-telepon, layanan telepon-ke-host, dan layanan konferensi video host-to-host multi-pihak. (Di sini, satu lagi host adalah perangkat IP yang terkoneksi Internet, termasuk PC, tablet, dan smartphone.) Skype diakuisisi oleh Microsoft di tahun 2011.

Untuk kedua suara dan video, klien Skype memiliki banyak codec yang berbeda, yang mampu mengkodekan media atau berbagai tingkat dan kualitas. Misalnya, kecepatan video untuk Skype telah diukur hingga 30 kbps untuk sesi berkualitas rendah hingga hampir 1 Mbps untuk sesi berkualitas tinggi [Zhang X 2012]. Biasanya, kualitas audio Skype lebih baik daripada "POTS" (Layanan Telepon Lama Biasa) berkualitas disediakan oleh sistem telepon kabel. (Skype codec biasanya sampel suara pada 16.000 sampel / detik atau lebih tinggi, yang menyediakan nada lebih kaya daripada POTS, yang sampelnya 8.000 / detik.).

Secara default, Skype mengirim paket audio dan video melalui UDP. Namun, paket kontrol dikirim melalui TCP, dan paket media juga lebih dari TCP ketika firewall memblokir aliran UDP. Skype menggunakan FEC untuk pemulihan kerugian untuk aliran suara dan video yang dikirim melalui UDP. Klien Skype juga mengadaptasi aliran audio dan video yang dikirimnya ke kondisi jaringan saat ini, dengan mengubah kualitas video dan overhead FEC [Zhang X 2012].

Skype menggunakan teknik P2P dalam sejumlah cara inovatif, dengan baik menggambarkan bagaimana P2P dapat digunakan dalam aplikasi yang melampaui distribusi konten dan berbagi file.



Gambar 9.7 *Skype Peer*

D. Protocols for Real-Time Conversational Applications / Protokol untuk Aplikasi Percakapan Real-Time

Dengan standar yang sesuai untuk percakapan Real Time aplikasi, perusahaan independen menciptakan produk baru yang saling beroperasi satu sama lain. Pada bagian ini kita menguji RTP dan SIP untuk aplikasi percakapan waktu-nyata. Kedua standar menikmati penerapan luas dalam produk industri.

1. RTP

Sebagian besar aplikasi jaringan multimedia dapat menggunakan urutan angka dan cap waktu, akan lebih mudah untuk memiliki struktur paket standar yang mencakup bidang untuk data audio / video, nomor urutan, dan stempel waktu, serta bidang lain yang berpotensi bermanfaat. RTP, didefinisikan dalam RFC 3550, adalah standar seperti itu. RTP dapat digunakan untuk mengangkut format umum seperti PCM, ACC, dan MP3 untuk suara dan MPEG dan H.263 untuk video. Ini juga dapat digunakan untuk mengangkut format suara dan video hak milik. Hari ini, RTP menikmati implementasi luas di banyak produk dan prototipe penelitian. Ini juga melengkapi protokol interaktif penting real time lainnya, seperti SIP.

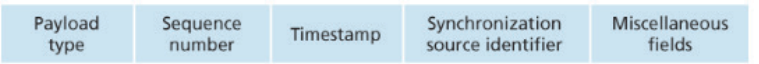
› Dasar RTP

RTP biasanya berjalan di atas UDP. Sisi pengiriman merangkum potongan media dalam paket RTP, kemudian merangkum paket dalam segmen UDP, dan kemudian menyerahkan segmen tersebut ke IP. Sisi penerima mengekstrak paket RTP dari segmen UDP, lalu mengekstrak potongan media dari paket RTP, dan kemudian meneruskannya ke pemutar media untuk decoding dan rendering.

Aplikasi mengekstrak audio chunk dari paket RTP dan menggunakan bidang header dari paket RTP untuk memecahkan kode dengan benar dan memutar audio chunk. Harus ditekankan bahwa RTP tidak menyediakan mekanisme untuk memastikan pengiriman data yang tepat waktu atau memberikan jaminan kualitas layanan lainnya (QoS); bahkan tidak menjamin pengiriman paket atau mencegah pengiriman paket yang tidak sesuai pesanan. Memang, enkapsulasi RTP hanya terlihat di

sistem akhir. Rute tidak membedakan antara datagram IP yang membawa paket RTP dan datagram IP yang tidak.

RTP memungkinkan setiap sumber (misalnya, kamera atau mikrofon) untuk ditugaskan aliran paket RTP independen. Misalnya, untuk konferensi video antara dua peserta, empat aliran RTP dapat dibuka dua aliran untuk mentransmisikan audio (satu di setiap arah) dan dua aliran untuk mentransmisikan video (sekali lagi, satu di setiap arah). Namun, banyak teknik penyandian populer termasuk MPEG 1 dan MPEG 2 bundel audio dan video ke dalam satu aliran tunggal selama proses penyandian. Ketika audio dan video dibundel oleh encoder, maka hanya satu aliran RTP yang dihasilkan di setiap arah.



Gambar 9.8 RTP header fields

› Bidang Header Paket RTP

Seperti yang ditunjukkan pada Gambar 9.8, empat bidang header paket RTP utama adalah tipe payload, sequencenumber, timestamp, dan bidang pengidentifikasi sumber. Bidang tipe payload dalam paket RTP adalah 7 bit. Untuk aliran audio, bidang jenis muatan digunakan untuk menunjukkan jenis pengkodean audio (misalnya, PCM, modulasi delta adaptif, pengkodean prediksi linear) yang digunakan. Jika pengirim memutuskan untuk mengubah pengkodean di tengah-tengah asesi, pengirim dapat menginformasikan penerima perubahan melalui bidang jenis muatan ini.

Untuk aliran video, jenis payload digunakan untuk menunjukkan jenis pengkodean video (misalnya, motionJPEG, MPEG 1, MPEG 2, H.261). Sekali lagi, pengirim dapat mengubah pengkodean video dengan cepat selama sesi berlangsung. Tabel 9.3 mencantumkan beberapa jenis payload video yang saat ini didukung oleh RTP. Bidang-bidang penting lainnya adalah sebagai berikut :

1) **Kolom nomor urut.** Bidang nomor urut panjangnya 16 bit. Peningkatan urutan nomor oleh satu untuk setiap paket RTP yang dikirim, dan dapat digunakan oleh penerima untuk

mendeteksi hilangnya paket dan untuk mengembalikan urutan paket.

2) **Bidang cap waktu.** Bidang cap waktu panjangnya 32 bit. Ini mencerminkan pengambilan sampel byte pertama dalam paket data RTP.

3) **Synchronization source identifier (SSRC)**. Kolom SSRC adalah 32 bit. Ini mengidentifikasi sumber aliran RTP. Biasanya, setiap aliran dalam sesi RTP memiliki SSRC yang berbeda. SSRC bukan alamat IP pengirim, melainkan nomor yang diberikan sumber secara acak ketika aliran baru dimulai. Probabilitas bahwa dua aliran mendapatkan SSRC yang sama sangat kecil. Jika ini terjadi, kedua sumber memilih nilai SSRC baru.

Tabel 9.2 Jenis Muatan Audio yang didukung oleh RTP

Nomor Jenis-Muatan	Format Audio	Tingkat Sampling	Nilai
0	PCM μ-law	8 kHz	64 kbps
1	1016	8 kHz	4.8 kbps
3	GSM	8 kHz	13 kbps
7	LPC	8 kHz	2.4 kbps
9	G.722	16 kHz	46-64 kbps
14	Audio MPEG	90 kHz	-
15	G.728	8 kHz	16 kbps

Tabel 9.3 Beberapa jenis payload video yang didukung oleh RTP

Nomor Jenis-Muatan	Format Vidio
26	Motion JPEG (Gerakan JPEG/GIF)
31	H.264
32	Vidio MPEG 1
33	Vidio MPEG 2

2. SIP

Session Initiation Protocol (SIP), didefinisikan dalam [RFC 3261; RFC 5411], adalah protokol terbuka dan ringan yang melakukan hal berikut:

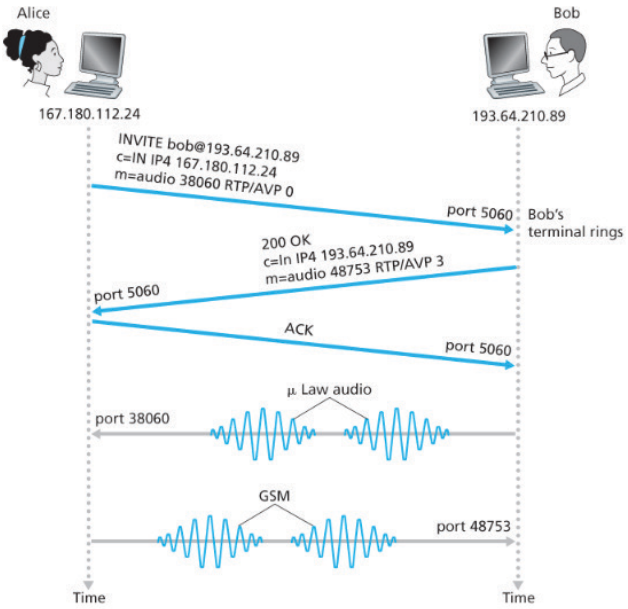
a. membuat panggilan antara penelepon dan callee melalui jaringan IP. Hal ini memungkinkan para peserta untuk menyetujui pengkodean media. Ini juga memungkinkan peserta untuk mengakhiri panggilan.

b. penelepon untuk menentukan alamat IP saat ini dari callee. Pengguna tidak memiliki satu alamat IP tetap karena mereka dapat diberi alamat secara dinamis (menggunakan DHCP) dan karena mereka mungkin memiliki beberapa perangkat IP, masing-masing dengan alamat IP yang berbeda.

c. manajemen panggilan, seperti menambahkan aliran media baru selama panggilan, mengubah pengodean selama panggilan, mengundang peserta baru selama panggilan, mentransfer panggilan, dan menahan panggilan.

› Menyiapkan Panggilan ke Alamat IP yang Dikenal

Untuk memahami esensi SIP, yang terbaik adalah melihat contoh konkret. Dalam contoh ini, Alice di PC-nya dan dia ingin memanggil Bob, yang juga bekerja di PC-nya. PC Alice dan Bob sama-sama dilengkapi dengan perangkat lunak berbasis SIP untuk melakukan dan menerima panggilan telepon. Dalam contoh awal ini, kita berasumsi bahwa Alice mengetahui alamat IP PC Bob. Gambar 9.9 menggambarkan proses pembentukan panggilan SIP.



Gambar 9.9 Pembuatan Panggilan SIP Ketika Alice Mengetahui Alamat IP Bob

Pada Gambar 9.9, kita melihat bahwa sesi SIP dimulai ketika Alice mengirimi Bob pesan undangan, yang mencakup pesan permintaan HTTP. Pesan undangan ini dikirim melalui UDP ke port5060 yang terkenal untuk SIP. (Pesan SIP juga dapat dikirim melalui TCP.) Pesan undangan termasuk pengidentifikasi untuk Bob (bob@193.64.210.89), indikasi alamat IP Alice saat ini, indikasi bahwa Alicedesires ingin menerima audio, yang akan dikodekan dalam format AVP 0 (PCM disandikan μ-law) dienkapsulasi dalam RTP, dan indikasi bahwa ia ingin menerima paket RTP di port 38060. Setelah menerima pesan INVITE Alice, Bob mengirim pesan respons SIP, yang menyerupai pesan respons HTTP.

Pesan SIP respons ini juga dikirim ke port SIP 5060. Respons Bob termasuk 200 OK dan juga indikasi alamat IP-nya, penyandian yang diinginkan dan penerimaan paketisasi yang diinginkan, dan nomor portnya ke mana paket audio harus dikirim. Perhatikan bahwa dalam contoh ini Alice dan Bob akan menggunakan mekanisme pengkodean audio yang berbeda: Alice diminta untuk

menyandikan heraudio dengan GSM sedangkan Bob diminta untuk menyandikan audionya dengan PCM μ-law. Setelah menerima respons Bob, Alice mengirimkan pesan pengakuan SIP kepada Bob. Setelah transaksi SIP ini, Bob dan Alicecan berbicara. (Untuk kenyamanan visual, Gambar 9.9 menunjukkan Alice berbicara setelah Bob, tetapi sebenarnya mereka tidak akan berbicara secara bersamaan.) Bob akan menyandikan dan mengemas audio sesuai permintaan dan mengirim paket audio ke nomor port 38060 di alamat IP 167.180.112.24. Alice juga akan menyandikan dan mengemas audio seperti yang diminta dan mengirim paket audio ke nomor port 48753 di alamat IP 193.64.210.89.

› Alamat SIP

Ketika perangkat SIP Alice mengirim pesan undangan, pesan akan menyertakan alamat seperti email ini; infrastruktur SIP kemudian akan merutekan pesan ke perangkat IP yang sedang digunakan Bob (seperti yang akan kita bahas di bawah). Bentuk lain yang mungkin untuk alamat SIP dapat berupa nomor telepon lawas Bob atau hanya nama depan / tengah / belakang Bob (dengan asumsi itu unik).

› Pesan SIP

Berikut adalah pesan undangan SIP, bersama dengan beberapa baris tajuk umum. Mari kita lagi berasumsi bahwa Alice ingin memulai panggilan VoIP ke Bob, dan kali ini Alice hanya tahu alamat SIP Bob, bob @ domain.com, dan tidak tahu alamat IP perangkat yang sedang digunakan Bob. Pesan selanjutnya mungkin terlihat seperti ini:

```
INVITE sip:bob@domain.com SIP/20
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp

Content-Length: 885c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Baris undangan menyertakan versi SIP, seperti halnya pesan permintaan HTTP. Setiap kali SIPmessage melewati perangkat SIP (termasuk perangkat yang berasal pesan), itu melampirkan

header Via, yang menunjukkan alamat IP perangkat. (Kita akan segera melihat bahwa pesan undangan biasa melewati banyak perangkat SIP sebelum mencapai aplikasi SIP callee.).

› Terjemahan Nama dan Lokasi Pengguna

Jadi sekarang mari kita anggap bahwa Alice hanya tahu alamat surel Bob, bob@domain.com, dan bahwa alamat yang sama ini digunakan untuk panggilan berbasis SIP. Dalam hal ini, Alice perlu mendapatkan alamat IP perangkat yang saat ini digunakan pengguna bob@domain.com. Untuk mengetahuinya, Alice membuat pesan undangan yang dimulai dengan undangan bob@ domain.com SIP / 2.0 dan mengirimkan pesan ini ke proxy SIP.

Setiap kali Bob beralih ke perangkat SIP baru, perangkat baru mengirim pesan register baru, yang menunjukkan alamat IP baru. Selain itu, jika Bob tetap berada di perangkat yang sama untuk periode waktu yang lama, perangkat akan mengirim pesan register penyegaran, yang menunjukkan bahwa alamat IP yang terakhir dikirim masih valid. (Dalam contoh di atas, refresh pesan yang perlu dikirim setiap 3600 detik untuk mempertahankan alamat di server registrar.) Perlu dicatat bahwa registrar analog dengan server nama otoritatif DNS: Server DNS menerjemahkan nama host tetap menjadi alamat IP tetap; pendaftar SIP menerjemahkan pengidentifikasi manusia tetap (misalnya, bob @ domain.com) ke alamat IP dinamis. Seringkali pendaftar SIP dan proksi SIP dijalankan di hosting yang sama.

Sebagai contoh, perhatikan Gambar 9.10, di mana jim@ umass.edu, saat ini bekerja pada 217.123.56.89, ingin memulai sesi Voice-over-IP (VoIP) dengan keith@upenn.edu, saat ini bekerja di197.87.54.21. Langkah-langkah berikut diambil

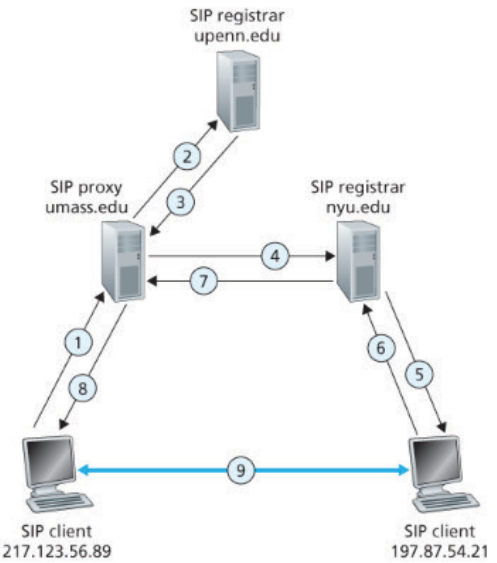


Figure 9.10 Session initiation, involving SIP proxies and registrars

Gambar 9.10 Inisiasi Sesi, yang melibatkan Proksi dan Pendaftar Sip

(1) Jim mengirim pesan undangan ke proxy SIP umass. (2) Proxy melakukan pencarian DNS pada registrasi SSIP upenn.edu (tidak ditampilkan dalam diagram) dan kemudian meneruskan pesan ke server registrar. (3) Karena keith@upenn.edu tidak lagi terdaftar di registrasi upenn, upenn registrar sendsa redirect response, menandakan bahwa ia harus mencoba keith@nyu.edu. (4) Proksi umass mengirim pesan undangan ke pendaftar SIP NYU. (5) Pendaftar NYU mengetahui alamat IP keith@upenn. eduand meneruskan pesan undangan ke host 197.87.54.21, yang menjalankan klien SIP Keith. (6–8) Respons SIP dikirim kembali melalui pendaftar / proksi ke klien SIP di 217.123.56.89. (9) Penerbitan media langsung antara kedua klien. (Ada juga pesan pengakuan SIP, yang tidak ditampilkan.

E. Dukungan Jaringan untuk Multimedia

Jaminan Kualitas Layanan (QoS) Per-koneksi. Dengan jaminan QoS per-koneksi, setiap instansi aplikasi secara eksplisit mencadangkan bandwidth end-to-end dan karenanya memiliki kinerja end-to-end yang

terjamin. Jaminan keras berarti aplikasi akan menerima kualitas layanan yang diminta (QoS) dengan pasti.

1. **Dimensi Jaringan Usaha Terbaik**

   Pada dasarnya, kesulitan dalam mendukung aplikasi multimedia muncul dari persyaratan kinerja mereka yang ketat, delay paket end-to-end yang rendah, delay jitter, dan loss dan fakta bahwa paket delay, delay jitter, dan kehilangan terjadi ketika jaringan menjadi padat.

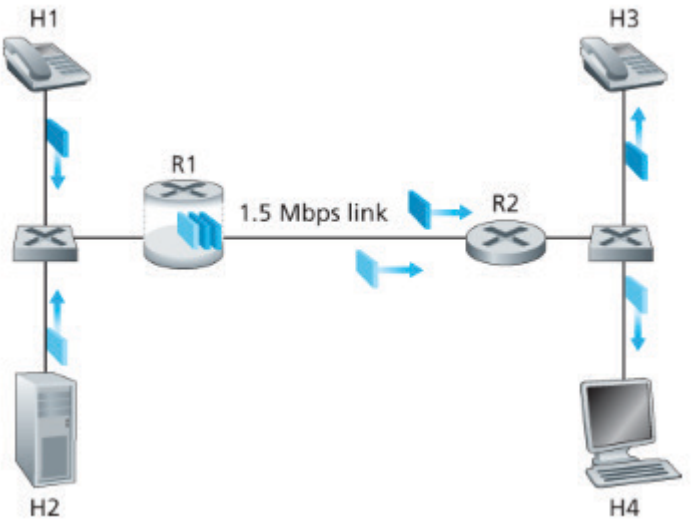
2. **Menyediakan Berbagai Kelas Layanan**

   Mungkin peningkatan paling sederhana untuk layanan upaya terbaik satu ukuran untuk semua upaya di Internet saat ini adalah untuk membagi lalu lintas ke dalam kelas, dan menyediakan tingkat layanan yang berbeda untuk berbagai kelas lalu lintas ini. Sebagai contoh, ISP mungkin ingin memberikan kelas layanan yang lebih tinggi untuk trafik konferensi Voice-over-IP Voice-over-IP sensitif (dan membebankan biaya lebih untuk layanan ini!) Daripada lalu lintas elastis seperti e-mail atau HTTP. Alternatifnya, ISP mungkin hanya ingin memberikan kualitas layanan yang lebih tinggi kepada pelanggan yang bersedia membayar lebih untuk layanan yang ditingkatkan.

   Perancang Internet awal jelas memiliki gagasan tentang berbagai kelas layanan dalam pikiran. Ingat bidang tipe layanan (ToS) di header IPv4 yang dibahas di Bab 4. IEN123 [ISI 1979] menjelaskan bidang Too juga hadir dalam leluhur datagram IPv4 sebagai berikut: "Jenis Layanan [bidang] memberikan indikasi parameter abstrak dari kualitas layanan yang diinginkan. Parameter ini dapat digunakan untuk memandu pemilihan parameter layanan aktual saat mengirimkan datagram melalui jaringan tertentu.

   Gambar 9.11 memperlihatkan skenario jaringan sederhana di mana dua aliran paket aplikasi berasal dari HostsH1 dan H2 pada satu LAN dan diperuntukkan bagi Hosts H3 dan H4 pada LAN lain. Perute pada duaLAN dihubungkan oleh tautan 1,5 Mbps. Mari kita asumsikan kecepatan LAN secara signifikan lebih tinggi dari 1.5Mbps, dan fokus pada antrian output router R1; di sinilah penundaan paket dan kehilangan paket akan terjadi jika laju pengiriman agregat H1 dan H2 melebihi 1,5 Mbps. Mari kita anggap lebih lanjut bahwa aplikasi 1 Mbpsaudio (misalnya, panggilan audio berkualitas CD) berbagi 1,5

Mbps menghubungkan antara R1 dan R2 dengan aplikasi penelusuran Web HTTP yang mengunduh Halaman Web dari H2 ke H4.



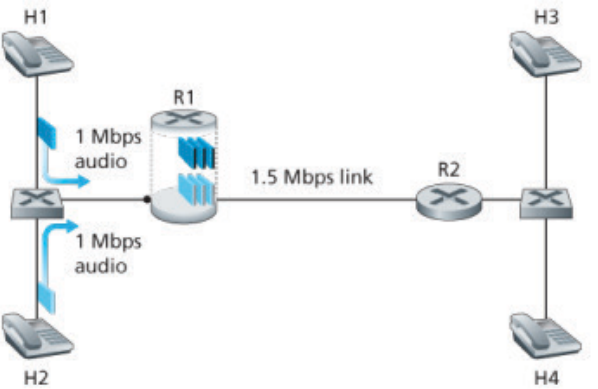
Gambar 9.11 Aplikasi audio dan HTTP yang bersaing

Di Internet upaya terbaik, paket audio dan HTTP dicampur dalam antrian output di R1 dan (biasanya) ditransmisikan dalam urutan pertama-masuk-pertama-keluar (FIFO). Dalam skenario ini, ledakan paket dari Web server berpotensi mengisi antrian, menyebabkan paket audio IP tertunda berlebihan atau kehilangan buffer overeto di R1. Bagaimana kita memecahkan masalah potensial ini? Mengingat bahwa aplikasi penjelajahan Web HTTP tidak memiliki batasan waktu, intuisi kita mungkin untuk memberikan prioritas yang ketat pada paket audio di R1. Di bawah disiplin penjadwalan prioritas yang ketat, paket audio dalam buffer output R1 akan selalu ditransmisikan sebelum paket HTTP apa pun di buffer output R1. Tautan dari R1 ke R2 akan terlihat seperti tautan khusus 1,5 Mbps ke lalu lintas audio, dengan lalu lintas HTTP menggunakan tautan R1-ke-R2 saja ketika tidak ada lalu lintas audio yang antri. Agar R1 membedakan antara antrian paket audio dan HTTP, setiap paket harus ditandai sebagai milik salah satu dari dua kelas lalu lintas ini. Ini adalah tujuan awal bidang tipe-layanan (ToS) di IPv4. Sejelas ini tampaknya, ini adalah wawasan pertama kita tentang mekanisme yang diperlukan untuk menyediakan beberapa kelas lalu lintas

3. Per-Connection Quality-of-Service (QoS)

Pada bagian sebelumnya, kita telah melihat bahwa penandaan dan pemolisian paket, isolasi lalu lintas, dan penjadwalan tautan-tingkat dapat memberikan satu kelas layanan dengan kinerja yang lebih baik daripada yang lain. Di bawah disiplin penjadwalan sertifikasi, seperti penjadwalan prioritas, kelas lalu lintas yang lebih rendah pada dasarnya "tidak terlihat" untuk kelas lalu lintas dengan prioritas tertinggi.

Mari kita kembali ke skenario kita dari Bagian 9.5.2 dan pertimbangkan dua aplikasi audio 1 Mbps mentransmisikan paket mereka melalui tautan 1,5 Mbps, seperti yang ditunjukkan pada Gambar 9.11. Kecepatan data gabungan dari dua aliran (2 Mbps) melebihi kapasitas tautan. Bahkan dengan klasifikasi dan penandaan, isolasi aliran, dan pembagian bandwidth yang tidak digunakan (yang tidak ada), ini jelas merupakan proposisi yang hilang.



Gambar 9.12 Dua aplikasi audio yang bersaing kelebihan tautan R1-ke-R2

waktu yang sama. Jika kedua aplikasi berbagi bandwidth secara sama, setiap aplikasi akan kehilangan 25 persen dari paket yang dikirimkan. Ini adalah QoS yang sangat rendah sehingga kedua aplikasi audio sama sekali tidak dapat digunakan; tidak perlu bahkan mengirim paket audio apa pun di tempat pertama.

Contoh motivasi kita pada Gambar 9.11 menggaris bawahi perlunya beberapa mekanisme jaringan baru dan protokol jika panggilan (aliran ujung ke ujung) harus dijamin dengan kualitas layanan yang diberikan begitu dimulai.

DAFTAR PUSTAKA

[Alizadeh 2010] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, M. Sridharan. "Data center TCP (DCTCP)," ACM SIGCOMM 2010 Conference, ACM, New York, NY, USA, pp. 63–74.

[Black 1995] U. Black, ATM Volume I: Foundation for Broadband Networks, Prentice Hall, 1995.

[Chen 2011] Y. Chen, S. Jain, V. K. Adhikari, Z. Zhang, "Characterizing Roles of Front-End Servers in End-to-End Performance of Dynamic Content Distribution," Proc. 2011 ACM Internet Measurement Conference (Berlin, Germany, Nov. 2011).

[Chiu 1989] D. Chiu, R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," Computer Networks and ISDN Systems, Vol. 17, No. 1, pp. 1–14. http://www.cs.wustl.edu/~jain/papers/cong_av.html

[Cisco 2015] Cisco Visual Networking Index: Forecast and Methodology, 2014–2019, White Paper, 2015

[Fauzan 2014] Fauzan Prasetyo, Supeno Djanali, "Optimasi Pengiriman Pesan pada Manet Protocol Routing Optimized Link State Routing (Olsr) dengan Menggunakan Evolutionary Algorithm", Jurnal Manajemen Informatika, Volume 03 Nomor 01 Tahun 2014 35-40

[Gude 2008] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an Operating System for

Networks," ACM SIGCOMM Computer Communication Review, July 2008

[ICANN 2016] The Internet Corporation for Assigned Names and Numbers homepage, http://www.icann.org

[IEEE 802.15.4 2012] IEEE 802.15 WPAN Task Group 4, http://www.ieee802.org/15/pub/TG4.html

[ISI 1979] Information Sciences Institute, "DoD Standard Internet Protocol," Internet Engineering Note 123 (Dec. 1979), http://www.isi.edu/in-notes/ien/ ien123.txt

[Jacobson 1988] V. Jacobson, "Congestion Avoidance and Control," Proc. 1988 ACM SIGCOMM (Stanford, CA, Aug. 1988), pp. 314–329

[Jain 1989] R. Jain, "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks," ACM SIGCOMM Computer Communications Review, Vol. 19, No. 5 (1989), pp. 56–71.

[Kreutz 2015] D. Kreutz, F.M.V. Ramos, P. Esteves Verissimo, C. Rothenberg, S. Azodolmolky, S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, Vol. 103, No. 1 (Jan. 2015), pp. 14-76. This paper is also being updated at https://github.com/SDN-Survey/latex/wiki

[Pathak 2010] A. Pathak, Y. A. Wang, C. Huang, A. Greenberg, Y. C. Hu, J. Li, K. W. Ross, "Measuring and Evaluating TCP Splitting for Cloud Services," Passive and Active Measurement (PAM) Conference (Zurich, 2010).

[Perkins 1998b] C. Perkins, Mobile IP: Design Principles and Practice, Addison-Wesley, Reading, MA, 1998.

[Ramakrishnan 1990] K. K. Ramakrishnan, R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," ACM Transactions on Computer Systems, Vol. 8, No. 2 (May 1990), pp. 158–181.

[Ramjee 1994] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," Proc. 1994 IEEE INFOCOM.

[RFC 1122] R. Braden, "Requirements for Internet Hosts—Communication Layers," RFC 1122, Oct. 1989

[RFC 1584] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Mar. 1994.

[RFC 1700] J. Reynolds, J. Postel, "Assigned Numbers," RFC 1700, Oct. 1994

[RFC 1930] J. Hawkinson, T. Bates, "Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)," RFC 1930, Mar. 1996.

[RFC 1945] T. Berners-Lee, R. Fielding, H. Frystyk, "Hypertext Transfer Protocol—HTTP/1.0," RFC 1945, May 1996.

[RFC 2018] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, "TCP Selective Acknowledgment Options," RFC 2018, Oct. 1996.

[RFC 2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, R. Fielding, "Hypertext Transfer Protocol—HTTP/1.1," RFC 2616, June 1999.

[RFC 3168] K. Ramakrishnan, S. Floyd, D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168, Sept. 2001

[RFC 3232] J. Reynolds, "Assigned Numbers: RFC 1700 Is Replaced by an On-line Database," RFC 3232, Jan. 2002.

[RFC 3261] J. Rosenberg, H. Schulzrinne, G. Carmarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, July 2002

[RFC 3416] R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)," Dec. 2002

[RFC 4271] Y. Rekhter, T. Li, S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2006

[RFC 4340] E. Kohler, M. Handley, S. Floyd, "Datagram Congestion Control Protocol (DCCP)," RFC 4340, Mar. 2006

[RFC 5411] J Rosenberg, "A Hitchhiker's Guide to the Session Initiation Protocol (SIP)," RFC 5411, Feb. 2009

[RFC 5681] M. Allman, V. Paxson, E. Blanton, "TCP Congestion Control," RFC 5681, Sept. 2009

[RFC 6265] A Barth, "HTTP State Management Mechanism," RFC 6265, Apr. 2011

[RFC 768] J. Postel, "User Datagram Protocol," RFC 768, Aug. 1980

[Sauter 2014] M. Sauter, From GSM to LTE-Advanced, John Wiley and Sons, 2014

[Schwartz 1982] M. Schwartz, "Performance Analysis of the SNA Virtual Route Pacing Control," IEEE Transactions on Communications, Vol. 30, No. 1 (Jan. 1982), pp. 172–184

[Shacham 1990] N. Shacham, P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management," Proc. 1990 IEEE INFOCOM (San Francisco, CA, Apr. 1990), pp. 124–131

[Skoudis 2006] E. Skoudis, T. Liston, Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses (2nd Edition), Prentice Hall, 2006

[Stone 1998] J. Stone, M. Greenwald, C. Partridge, J. Hughes, "Performance of Checksums and CRC's Over Real Data," IEEE/ACM Transactions on Networking, Vol. 6, No. 5 (Oct. 1998), pp. 529–543.

[Stone 2000] J. Stone, C. Partridge, "When Reality and the Checksum Disagree," Proc. 2000 ACM SIGCOMM (Stockholm, Sweden, Aug. 2000)

[Zhang X 2102] X. Zhang, Y. Xu, Y. Liu, Z. Guo, Y. Wang, "Profiling Skype Video Calls: Rate Control and Video Quality," IEEE INFOCOM (Mar. 2012)

JARINGAN KOMPUTER

Untuk Pemula

Dengan penuh kebahagiaan, kami mempersembahkan buku ini untuk para profesional dan pelajar bidang teknologi informasi. Dalam buku ini, kami akan membahas tentang dunia jaringan komputer, sebuah bidang yang sangat penting dan berkembang pesat dalam era digital saat ini.

Jaringan komputer merupakan dasar dari segala aktivitas online dan memegang peran yang sangat penting dalam mengatur dan menyediakan akses informasi bagi masyarakat. Oleh karena itu, penting bagi kita untuk memahami bagaimana jaringan komputer bekerja dan bagaimana mengatasi masalah yang muncul dalam implementasi jaringan.

Buku ini dirancang untuk memberikan pemahaman dasar tentang jaringan komputer dan membahas topik-topik yang penting dalam bidang ini, seperti protokol jaringan, keamanan jaringan, dan implementasi jaringan. Kami berharap bahwa buku ini dapat menjadi sumber informasi yang berguna bagi para pemula dan profesional yang ingin memperdalam pengetahuan mereka tentang jaringan komputer.

Kami berharap buku ini dapat membantu Anda memahami jaringan komputer dengan lebih baik dan membantu Anda dalam mempersiapkan diri untuk menghadapi tantangan dan peluang di bidang teknologi informasi. Selamat membaca.

Penerbit litnus.
literasinusantaraofficial@gmail.com
www.penerbitlitnus.co.id
@litnuspenerbit
literasinusantara_
085755971589

Sains +17
ISBN 978-623-8227-44-0
9 786238 227440