Weekly Log
Week 3

Jonathan Filbert
JARKOMDAT - A
1806191692

Topic: Application Layer: Network Application and HTTP

1. Network Application

Network application adalah aplikasi yang memiliki beberapa sifat berikut:
- Terdiri dari client dan server
- Berkomunikasi melalui jaringan
- Membuka *socket* di host masing-masing untuk berkomunikasi

Network Application pasti terdiri dari minimal 2 komponen host berbeda, yaitu client dan server. Ada 2 tipe arsitektur network application:

1. Client Server
   Client server adalah arsitektur dimana ada 2 jenis host berbeda yang dimana satu adalah client, dan satunya adalah server. Client dan server boleh lebih dari satu tentu saja.
   Tugas client adalah yang berhubungan langsung dan berinteraksi dengan user, sedangkan server menerima data yang dikirim oleh client, dan melakukan pemrosesan data, dan kemudian dikirim kembali ke client untuk ditampilkan dan diberitahu kepada pengguna.
   Client memiliki IP yang dinamis, dan tidak harus selalu online.
   Server adalah host adalah menerima data dari client dan kemudian melakukan pemrosesan data, seperti melakukan query database, melakukan komputasi,dll kemudian hasilnya dikirim kembali ke client. Server harus online 24/7 karena request dari client bisa masuk kapan saja, serta server harus memiliki IP yang statis yang artinya tidak boleh berubah.
2. Peer to peer
   Peer to peer adalah arsitektur yang memungkinkan sebuah host menjadi client dan server disaat yang bersamaan. Sebuah host akan berinteraksi dengan host lain, yang dimana host itu akan merequest data dan disaat yang sama menerima data dari host lain. Contoh teknologi peer to peer adalah ***torrenting***.

Process adalah program yang berjalan disebuah host yang dimana diatur oleh operating system. Dalam client server, process yang berjalan pada client adalah process yang akan mengirimkan data kepada server, sedangkan process pada server adalah process yang menunggu untuk dikirimkan data dari client untuk kemudian diproses dan mengirim hasilnya kepada client.

Socket adalah sebuah komponen yang terletak diantara application dan transport layer yang dimana berperan sebagai pintu tempat keluar masuk data yang ingin dikirim / sudah diterima ke application layer.

Port adalah komponen pada OS yang dimana dapat diisi dan mengandung proses yang sedang berjalan. Kadang-kadang proses kita ingin mengirim dan menerima data, sehingga proses yang kita jalankan harus memiliki identifier unik sendiri, untuk itu kita perlu menjalankan sebuah proses didalam sebuah port unik supaya request yang dibuat dan diterima dapat dibedakan dengan proses lain yang sedang berjalan.

Transport Protocol

Transport protocol adalah protokol pengiriman data dari/kepada application layer di transport layer. Ada 2 protokol yang sering digunakan, yaitu:

1. TCP
   TCP adalah protokol untuk pengiriman data yang tidak mentolerir packet-loss, sehingga semua packet yang dikirimkan menggunakan TCP akan dicek dan melalui berbagai proses kontrol yang berat supaya dapat dipastikan ketika sampai ke tujuan, semua packet 100% sampai.
   TCP dipakai untuk mengirim data-data yang tidak boleh mengalami packet loss, contohnya adalah email (SMTP) dan pengiriman file (FTP).
2. UDP
   UDP adalah protokol pengiriman data yang mentolerir packet loss, sehingga bila terjadi packet loss di UDP, akan dibiarkan. Tetapi karena dia mentolerir packet loss, maka UDP tidak memiliki berbagai mekanisme control yang dimiliki oleh TCP, sehingga UDP bisa lebih cepat.

Jonathan Filbert
JARKOMDAT - A
1806191692

UDP biasa dipakai untuk mengirim data-data yang boleh hilang, sehingga kadang fenomena kita streaming media dan ada packet hilang karena throughput internet kita kecil, atau jaringan kita sedang ramai disebut “buffering”. Contoh penggunaan UDP adalah media streaming.

TCP dan UDP adalah protokol yang tidak melalui enkripsi, untuk melakukan TCP yang lebih aman, maka digunakanlah **SSL** atau (Secure Socket Layer) untuk melakukan enkripsi pada TCP dan UDP yang keluar/masuk. Website yang sudah menggunakan SSL biasa memiliki https daripada http.

2. HTTP

HTTP atau HyperText Transfer Protocol adalah protokol yang berjalan di application layer yang dimana akan digunakan client untuk merequest dan digunakan server untuk mengirimkan response.

Sebelum memulai sebuah HTTP request, maka client harus membuka socketnya terlebih dahulu untuk memberi tahu kepada server socket mana untuk memberikan response. Pembukaan socket mengharuskan client mengirim sebuah TCP request yang kemudian akan dibalas dengan TCP response oleh server, setelah TCP dan port telah terbuka, maka client baru bisa mengirimkan HTTP request.
HTTP memiliki 2 jenis, yaitu:

1. Persistent
   Setelah port dibuka oleh TCP request pertama kali, maka HTTP Persist akan membiarkan port terbuka, sehingga HTTP request selanjutnya tidak perlu mengirimkan TCP untuk membuka port lagi.
2. Non-persistent
   Setelah port dibuka oleh TCP request pertama kali, setelah HTTP response sudah sampai, maka HTTP Non-persist akan menutup socket, sehingga bila client ingin mengirimkan HTTP request, maka harus mengirimkan TCP request untuk membuka socket kembali.

RTT atau **Round Trip Time** adalah waktu yang dibutuhkan sebuah packet untuk dikirim dari client ke server, dan kembali ke client dengan membawa response dari server.

Untuk HTTP non-persistent, maka **response time** dapat dihitung dengan 1 kali RTT (inisiasi TCP) + 1 RTT (HTTP request-response) + file transmission time, untuk HTTP persistent, maka response time nya adalah 1RTT (HTTP Request-response) + File transmission time + 1RTT (inisiasi socket) diawal.

HTTP memiliki beberapa metode, berikut adalah beberapa metode HTTP:
1. GET
2. POST
3. PUT (HTTP/1.1)
4. DELETE (HTTP/1.1
5. PATCH (HTTP/1.1)

HTTP memiliki beberapa response code, berikut beberapa response code:
1. 200 - OK
2. 404 - Not Found
3. 403 - Forbidden
4. 405 - Not Allowed
5. 400 - Bad Request
6. 301 - Moved Permanently
7. 505 - HTTP version not supported
8. 418 - I’m a teapot