TUGAS JARINGAN KOMPUTER

LAYANAN DARI TCP DAN UDP PROTOCOL

DITULIS OLEH :

ADRIAN AJISMAN

09011281520133

SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2017

1. TCP

1.1 Pengertian TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol) adalah salah satu jenis protokol yang memungkinkan sekumpulan komputer untuk berkomunikasi dan bertukar data didalam suatu jaringan

1.2 Karakteristik TCP

UDP memiliki karakteristik-karakteristik berikut:

a. ***Connectionless*** (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak bertukar informasi.
b. ***Unreliable*** (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan acknowledgment. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi.
c. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP.
d. UDP menyediakan penghitungan checksum berukuran 16-bit terhadap keseluruhan pesan UDP.

Header TCP

Ukuran TCP header paling kecil (ketika tidak ada tambahan opsi TCP) adalah 20 byte. headerTCP-2

Nama field	Ukuran	Keterangan
Source Port	2 byte (16 bit)	Mengindikasikan sumber protokol lapisan aplikasi yang mengirimkan segmen TCP yang bersangkutan. Gabungan antara *field* **Source IP Address** dalam *header IP* dan *field* **Source Port** dalam *field header TCP* disebut juga sebagai **source *socket***, yang berarti sebuah alamat global dari mana segmen dikirimkan. Lihat juga Port TCP.
Destination Port	2 byte (16 bit)	Mengindikasikan tujuan protokol lapisan aplikasi yang menerima segmen TCP yang bersangkutan. Gabungan antara field Destination IP Address dalam header IP dan field Destination Port dalam field header TCP disebut juga sebagai ***socket* tujuan**, yang berarti sebuah alamat global ke mana segmen akan dikirimkan.
Sequence Number	4 byte (32 bit)	Mengindikasikan nomor urut dari oktet pertama dari data di dalam sebuah segmen TCP yang hendak dikirimkan. Field ini harus selalu diset, meskipun tidak ada data (payload) dalam segmen. Ketika memulai sebuah sesi koneksi TCP, segmen dengan flag SYN (Synchronization) diset ke nilai 1, field ini akan berisi nilai Initial Sequence Number (ISN). Hal ini berarti, oktet pertama dalam aliran byte (byte stream) dalam koneksi adalah ISN+1.
Acknowledgment Number	4 byte (32 bit)	Mengindikasikan nomor urut dari oktet selanjutnya dalam aliran byte yang diharapkan oleh untuk diterima oleh pengirim dari si penerima pada pengiriman selanjutnya. Acknowledgment number sangat dipentingkan bagi segmen-segmen TCP dengan flag ACK diset ke nilai 1.
Data Offset	4 bit	Mengindikasikan di mana data dalam segmen TCP dimulai. Field ini juga dapat berarti ukuran dari header TCP. Seperti halnya field **Header Length** dalam header IP, field ini merupakan angka dari word 32-bit dalam header TCP. Untuk sebuah segmen TCP terkecil (di mana tidak ada opsi TCP tambahan), field ini diatur ke nilai 0x5, yang berarti data dalam segmen TCP dimulai dari oktet ke 20 dilihat dari permulaan segmen TCP. Jika field Data Offset diset ke nilai maksimumnya (2^4=16) yakni 15, header TCP dengan ukuran terbesar dapat memiliki panjang hingga 60 byte.
Reserved	6 bit	Direservasikan untuk digunakan pada masa depan. Pengirim segmen TCP akan mengeset bit-bit ini ke dalam nilai 0.
Flags	6 bit	Mengindikasikan flag-flag TCP yang memang ada enam jumlahnya, yang terdiri atas: URG (Urgent), ACK (Acknowledgment), PSH (Push), RST (Reset), SYN (Synchronize), dan FIN (Finish).
Window	2 byte (16 bit)	Mengindikasikan jumlah byte yang tersedia yang dimiliki oleh buffer host penerima segmen yang bersangkutan. Buffer ini disebut sebagai Receive Buffer, digunakan untuk menyimpan byte stream yang datang. Dengan mengimbuhkan ukuran window ke setiap segmen, penerima segmen TCP memberitahukan kepada pengirim segmen berapa banyak data yang dapat dikirimkan dan disangga dengan sukses. Hal ini dilakukan agar si pengirim segmen tidak mengirimkan data lebih banyak dibandingkan ukuran Receive Buffer. Jika tidak ada tempat lagi di dalam Receive buffer, nilai dari field ini adalah 0. Dengan nilai 0, maka si pengirim tidak akan dapat mengirimkan segmen lagi ke penerima hingga nilai field ini berubah (bukan 0). Tujuan hal ini adalah untuk mengatur lalu lintas data atau *flow control*.
Checksum	2 byte (16 bit)	Mampu melakukan pengecekan integritas segmen TCP (*header*-nya dan *payload*-nya). Nilai field Checksum akan diatur ke nilai 0 selama proses kalkulasi checksum.
Urgent Pointer	2 byte (16 bit)	Menandakan lokasi data yang dianggap "urgent" dalam segmen.
Options	4 byte (32 bit)	Berfungsi sebagai penampung beberapa opsi tambahan TCP. Setiap opsi TCP akan memakan ruangan 32 bit, sehingga ukuran header TCP dapat diindikasikan dengan menggunakan field Data offset.

(Sumber : id.wikipedia.org)

Port TCP

Port TCP mampu mengindikasikan sebuah lokasi tertentu untuk menyampaikan segmen-segmen TCP yang dikirimkan yang diidentifikasi dengan TCP Port Number.

1.3 Layanan TCP

Beberapa kegunaan dari TCP yaitu :

a. Menyediakan komunikasi logika antar proses aplikasi yang berjalan pada host yang berbeda
b. Protokol transport berjalan pada end systems
c. Pengiriman file (file transfer). File Transfer Protokol (FTP) memungkinkan pengguna komputer yg satu untuk dapat mengirim ataupun menerima file ke komputer jaringan. Karena masalah keamanan data, maka FTP seringkali memerlukan nama pengguna (username) dan password, meskipun banyak juga FTP yg dapat diakses melalui anonymous, lias tidak berpassword. (lihat RFC 959 untuk spesifikasi FTP)

d. Remote login. Network terminal Protokol (telnet) memungkinkan pengguna komputer dapat melakukan log in ke dalam suatu komputer didalam suatu jaringan. Jadi hal ini berarti bahwa pengguna menggunakan komputernya sebagai perpanjangan tangan dari komputer jaringan tersebut.(lihat RFC 854 dan 855 untuk spesifikasi telnet lebih lanjut)
e. Computer mail. Digunakan untuk menerapkan sistem elektronik mail.
f. Network File System (NFS). Pelayanan akses file-file jarak jauh yg memungkinkan klien-klien untuk mengakses file-file pada komputer jaringan jarak jauh walaupun file tersebut disimpan secara lokal. (lihat RFC 1001 dan 1002 untuk keterangan lebih lanjut)
g. Remote execution. Memungkinkan pengguna komputer untuk menjalankan suatu program didalam komputer yg berbeda. Biasanya berguna jika pengguna menggunakan komputer yg terbatas, sedangkan ia memerlukan sumber yg banyak dalam suatu system komputer. Ada beberapa jenis remote execution, ada yg berupa perintah-perintah dasar saja, yaitu yg dapat dijalankan dalam system komputer yg sama dan ada pula yg menggunakan "prosedure remote call system", yg memungkinkan program untuk memanggil subroutine yg akan dijalankan di system komputer yg berbeda. (sebagai contoh dalam Berkeley UNIX ada perintah "rsh" dan "rexec")
h. Name servers. Nama database alamat yg digunakan pada internet (lihat RFC 822 dan 823 yg menjelaskan mengenai penggunaan protokol name server yg bertujuan untuk menentukan nama host di internet.)

Aplikasi yang Menggunakan TCP

a. World Wide Web
b. Archie
c. Wide Area Information Services (WAIS)
d. FAX di Internet

2. UDP

2.1 Pengertian UDP (User Datagram Protocol)

UDP (User Datagram Protocol) adalah salah satu protokol lapisan transport TCP/IP yang mendukung komunikasi yang tidak handal (unreliable), tanpa koneksi antara host-host dalam jaringan yang menggunakan TCP/IP.

2.2 Karakteristik UDP

TCP memiliki karakteristik sebagai berikut:

a. Berorientasi sambungan (*connection-oriented*): Sebelum data dapat ditransmisikan antara dua host, dua proses yang berjalan pada lapisan aplikasi harus melakukan negosiasi untuk membuat sesi koneksi terlebih dahulu. Koneksi TCP ditutup dengan menggunakan proses terminasi koneksi TCP (TCP connection termination).

b. *Full-duplex*: Untuk setiap host TCP, koneksi yang terjadi antara dua host terdiri atas dua buah jalur, yakni jalur keluar dan jalur masuk.

c. Dapat diandalkan (*reliable*): Data yang dikirimkan ke sebuah koneksi TCP akan diurutkan dengan sebuah nomor urut paket dan akan mengharapkan paket *positive acknowledgment* dari penerima. Jika tidak ada paket Acknowledgment dari penerima, maka segmen TCP (protocol data unit dalam protokol TCP) akan ditransmisikan ulang.

d. *Byte stream*: TCP melihat data yang dikirimkan dan diterima melalui dua jalur masuk dan jalur keluar TCP sebagai sebuah *byte stream* yang berdekatan (kontigu). Nomor urut TCP dan nomor acknowlegment dalam setiap header TCP didefinisikan juga dalam bentuk byte.

e. Memiliki layanan *flow control*: Untuk mencegah data terlalu banyak dikirimkan pada satu waktu, yang akhirnya membuat "macet" jaringan internetwork IP, TCP mengimplementasikan layanan *flow control* yang dimiliki oleh pihak pengirim yang secara terus menerus memantau dan membatasi jumlah data yang dikirimkan pada satu waktu.

f. Melakukan segmentasi terhadap data yang datang dari lapisan aplikasi (dalam *DARPA Reference Model*)
g. Mengirimkan paket secara "*one-to-one*": hal ini karena memang TCP harus membuat sebuah sirkuit logis antara dua buah protokol lapisan aplikasi agar saling dapat berkomunikasi. TCP tidak menyediakan layanan pengiriman data secara *one-to-many*.

Header UDP

Header UDP diwujudkan sebagai sebuah header dengan 4 buah field memiliki ukuran yang tetap.

Field	Panjang	Keterangan
Source Port	16 bit (2 byte)	Digunakan untuk mengidentifikasikan sumber protokol lapisan aplikasi yang mengirimkan pesan UDP yang bersangkutan. Penggunaan field ini adalah opsional, dan jika tidak digunakan, akan diset ke angka 0. Beberapa protokol lapisan aplikasi dapat menggunakan nilai field ini dari pesan UDP yang masuk sebagai nilai field port tujuan (Destination Port, lihat baris selanjutnya) sebagai balasan untuk pesan tersebut.
Destination Port	16 bit (2 byte)	Digunakan untuk mengidentifikasikan tujuan protokol lapisan aplikasi yang menjadi tujuan pesan UDP yang bersangkutan. Dengan menggunakan kombinasi antara alamat IP dengan nilai dari field ini untuk membuat sebuah alamat yang signifikan untuk mengidentifikasikan proses yang berjalan dalam sebuah host tertentu yang dituju oleh pesan UDP yang bersangkutan.
Length	16 bit (2 byte)	Digunakan untuk mengindikasikan panjang pesan UDP (pesan UDP ditambah dengan header UDP) dalam satuan byte. Ukuran paling kecil adalah 8 byte (ukuran header UDP, ketika tidak ada isi pesan UDP), dan ukuran paling besar adalah 65515 bytes (65535 [2^{16}] -20 [ukuran header protokol IP]). Panjang maksimum aktual dari pesan UDP akan disesuaikan dengan menggunakan nilai Maximum Transmission Unit (MTU) dari saluran di mana pesan UDP dikirimkan. Field ini bersifat redundan (terulang-ulang). Panjang pesan UDP dapat dihitung dari field Length dalam header UDP dan field IP Header Length dalam header IP.
Checksum	16 bit (2 byte)	Berisi informasi pengecekan integritas dari pesan UDP yang dikirimkan (header UDP dan pesan UDP). Penggunaan field ini adalah opsional. Jika tidak digunakan, field ini akan bernilai 0.

(Sumber : id.wikipedia.org)

Port UDP

Seperti halnya TCP, UDP juga memiliki saluran untuk mengirimkan informasi antar host, yang disebut dengan UDP Port. Untuk menggunakan protokol UDP, sebuah aplikasi harus menyediakan alamat IP dan nomor UDP Port dari host yang dituju. Sebuah UDP port berfungsi sebagai sebuah multiplexed message queue, yang berarti bahwa UDP port tersebut dapat menerima beberapa pesan secara sekaligus. Setiap port diidentifikasi dengan nomor yang unik, seperti halnya TCP, tetapi meskipun begitu, UDP Port berbeda dengan TCP Port meskipun memiliki nomor port yang sama.

2.3 Layanan UDP

UDP sering digunakan dalam beberapa tugas berikut:

a. Protokol yang “ringan” (lightweight): Untuk menghemat sumber daya memori dan prosesor, beberapa protokol lapisan aplikasi membutuhkan penggunaan protokol yang ringan yang dapat melakukan fungsi-fungsi spesifik dengan saling bertukar pesan. Contoh dari protokol yang ringan adalah fungsi query nama dalam protokol lapisan aplikasi Domain Name System.

b. Protokol lapisan aplikasi yang mengimplementasikan layanan keandalan: Jika protokol lapisan aplikasi menyediakan layanan transfer data yang andal, maka kebutuhan terhadap keandalan yang ditawarkan oleh TCP pun menjadi tidak ada. Contoh dari protokol seperti ini adalah Trivial File Transfer Protocol (TFTP) dan Network File System (NFS)

c. Protokol yang tidak membutuhkan keandalan. Contoh protokol ini adalah protokol Routing Information Protocol (RIP).

d. Transmisi broadcast: Karena UDP merupakan protokol yang tidak perlu membuat koneksi terlebih dahulu dengan sebuah host tertentu, maka transmisi broadcast pun dimungkinkan. Sebuah protokol lapisan aplikasi dapat mengirimkan paket data ke beberapa tujuan dengan menggunakan alamat multicast atau broadcast. Hal ini kontras dengan protokol TCP yang hanya dapat mengirimkan transmisi one-to-one. Contoh: query nama dalam protokol NetBIOS Name Service.

Contoh protokol aplikasi yang menggunakan UDP :

a. DNS (Domain Name System) 53
b. SNMP, (Simple Network Management Protocol) 161, 162
c. TFTP (Trivial File Transfer Protocol) 69
d. unRPC port 111.

3. Perbedaan TCP dan UDP

Beberapa perbedaan dan TCP dan UDP antara lain :

No	TCP	UDP
1.	Beroperasi berdasarkan konsep koneksi.	Tidak berdasarkan konsep koneksi, jadi harus membuat kode sendiri.
2.	Jaminan pengiriman-penerimaan data akan reliable dan teratur.	Tidak ada jaminan bahwa pengiriman dan penerimaan data akan reliable dan teratur, sehingga paket data mungkin dapat kurang, terduplikat, atau bahkan tidak sampai sama sekali.
3.	Secara otomatis memecah data ke dalam paket-paket.	Pemecahan ke dalam paket-paket dan proses pengirimannya dilakukan secara manual.
4.	Tidak akan mengirimkan data terlalu cepat sehingga memberikan jaminan koneksi internet dapat menanganinya.	Harus membuat kepastian mengenai proses transfer data agar tidak terlalu cepat sehingga internet masih dapat menanganinya.
5.	Mudah untuk digunakan, transfer paket data seperti menulis dan membaca file.	Jika paket ada yang hilang, perlu dipikirkan di mana letak kesalahan yang terjadi dan mengirim ulang data yang diperlukan.

(Sumber : blog.klikstream)

No	TCP	UDP
1.	**Dapat diandalkan** Jika sambungan terputus ketika mengrim sebuah pesan maka server akan meminta bagian yang hilang. Jadi tidak akan terjadi data yang korup ketika mentransfer sebuah data.	**Tidak dapat diandalkan** Jika mengirimkan suatu pesan atau data, kita tidak akan tahu apakah sudah terkirim atau belum dan apakah sebagian dari pesan tersebut hilang atau tidak ketika proses pengiriman. Jadi akan ada kemungkinan terjadinya data yang korup.
2.	**Berurutan** Ketika mengrimkan dua pesan secara berurutan / satu demi satu. TCP akan mengirimkannya secara berurutan. Tidak perlu khawatir data tiba dengan urutan yang salah.	**Tidak berurutan** Ketika mengrimkan dua pesan secara berurutan / satu demi satu. Tidak dapat dipastikan data mana yang akan datang terlebih dahulu.
3.	**Berorientasi sambungan (connection-oriented)**Sebelum data dapat ditransmisikan antara dua host, dua proses yang berjalan pada lapisan aplikasi harus melakukan negosiasi untuk membuat sesi koneksi terlebih dahulu. Koneksi TCP ditutup dengan menggunakan proses terminasi koneksi TCP (TCP connection termination).	**Connectionless (tanpa koneksi)** Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak berukar informasi.

4.	**Ringan (Heavyweight)** Ketika tingkat level terendah dari TCP tercapai dalam urutan yang salah,permintaan pengiriman ulang data harus dikirm. dan bagian lainya harus dikembalikan semua. Sehingga membutuhkan proses untuk menyatukannya	**Ringan (Lightweight)** Tidak ada permintaan pesan, tidak ada trak koneksi dan yang lainnya, hanya menjalankan dan melupakannya. Ini berarti itu jauh lebih cepat dan kartu jaringan / OS hanya melakukan sedikit pekerjaan untuk menerjemahkan kembali data dari paket.
5.	**Streaming** Data /paket dibaca sebagai satu alur data. tanpa mengetahui batas setiap data berakhir dan data yang lain mulai. Ada kemungkinan beberapa paket data dibaca per satu panggilan data.	**Datagrams** Paket dikirim secara individu dan dijamin utuh ketika tiba. Satu paket dibaca per satu panggilan.
5.	**Contoh** World Wide Web (Apache TCP port 80), e-mail (SMTP TCP port 25 Postfix MTA), File Transfer Protocol (FTP port 21) and Secure Shell (OpenSSH port 22) etc.	**Contoh** Domain Name System (DNS UDP port 53), streaming media applications such as IPTV or movies, Voice over IP (VoIP), Trivial File Transfer Protocol (TFTP) and online multiplayer games etc

(Sumber : blog.klikstream)

4. Rujukan

http://blog.klikstream.co.id/2010/04/pengertian-dan-perbedaan-tcp-dan-udp.html

https://id.wikipedia.org/wiki/User_Datagram_Protocol

https://id.wikipedia.org/wiki/Transmission_Control_Protocol

http://irpantips4u.blogspot.co.id/2012/11/tcp-dan-udp-penjelasan-dan-perbedaannya.html

https://klikhost.com/perbedaan-tcp-dan-udp/