

MEMBANDINGKAN PROTOKOL UDP (*User Datagram Protokol*) & TCP (*Transmisson Control Protocol*)

Mardianto,*S.Kom,M.Cs*^{*}, Nur Lisdayanti^{**}, Jelvina Rizka^{***}, Nurul Rahmi Ramadhani^{****}

Program Studi Sistem Informasi, Universitas Sembilanbelas November, Kolaka, Sulawesi Tenggara, Indonesia

e-mail: nurlisdayanti17@gmail.com, jelvinarizka@gmail.com, nurulrhmiramadhani@gmail.com

Abstrak: Pemanfaatan layanan multimedia saat ini telah dilaksanakan secara meluas dalam berbagai tujuan. Karena perkembangannya yang pesat, maka diperlukan suatu aturan yang mampu meningkatkan kualitas dari layanan ini. Suatu protokol adalah sebuah aturan yang mendefinisikan beberapa fungsi yang ada dalam sebuah jaringan komputer, misalnya mengirim pesan, data, informasi, dan fungsi ini yang harus di penuhi oleh pengirim (*transametiter*) dan penerima (*receiver*) agar komunikasi dapat berlangsung dengan benar. Jurnal ini menggambarkan hasil perbandingan protokol *User Datagram Protocol* (UDP), dan *Transmission Control Protocol* (TCP).

Kata Kunci : *Protokol UDP, Protokol TCP*

COMPARE UDP (*User Datagram Protocol*) & TCP (*Transmission Control Protocol*) PROTOCOLS

Abstract: *Currently, the use of multimedia services has been carried out widely for various purposes. Due to its rapid development, a regulation is needed that can improve the quality of this service. A protocol is a rule that defines several functions that exist in a computer network, for example sending messages, data, information, and this function must be fulfilled by the sender (transameter) and receiver (receiver) so that communication can take place correctly. This journal describes the results of the comparison of the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) protocols.*

Keywords: *UDP Protocol, TCP Protocol*

1) PENDAHULUAN

Pada *internet protocol suite* , layer transport merupakan layer yang berada diatas layer network. Jika layer network berfungsi untuk mengatur transfer data antar end-system, maka fungsi dari layer transport ini adalah untuk mengatur transfer data antar proses. Proses yang terjadi pada saat transfer data ini bisa menggunakan berbagai macam protocol seperti ;UDP (*User Datagram Protocol*), TCP

(Transmission Control Protocol), DCCP (Diagram Congestion Control Protocol), SCTP (Stream Control Transmission Protocol), dan RSVP (Resource Reservation Protocol). Dimana tiap-tiap protocol tersebut memiliki karakteristik dan fungsi yang berbeda-beda. Diantara beberapa protokol pada layer tersebut yang paling sering digunakan adalah UDP dan TCP. Pada artikel ini akan dibahas tentang perbandingan kedua protocol tersebut.

2) METODE PENELITIAN

Dalam penelitian ini penulis menggunakan metode penelitian kualitatif eksperimen. Metode ini memperoleh pemahaman mendalam, mengembangkan teori, mendeskripsikan realitas, dan kompleksitas sosial.

3) HASIL DAN PEMBAHASAN

3.1 PROTOKOL UDP

UDP merupakan protocol internet yang mengutamakan kecepatan data. Protocol ini biasanya digunakan untuk streaming video ataupun fasilitas *real-time* yang lain. Oleh karena itu pada UDP ini tidak memerlukan adanya setup koneksi terlebih dahulu karena hal tersebut dapat menyebabkan adanya tambahan delay. Selain itu, protocol ini termasuk dalam protocol yang sederhana, artinya antara penerima dan pengirim tidak perlu menjaga session atau status koneksi, ukuran headernya juga sederhana. UDP ini juga tidak memerlukan *congestion control* (control kemacetan) pada koneksinya. Maksudnya adalah UDP dapat mengirimkan per segment tanpa dipengaruhi oleh kesibukan jaringan. Berikut karakteristik lebih lanjut dari Protokol UDP :

- a) *Connectionless*, maksudnya adalah tidak ada aktifitas handshaking antara UDP dan penerimanya saat akan dilakukan pengiriman data sehingga data tersebut dikirim melalui jaringan dan mencapai ke computer tujuan tanpa membuat suatu koneksi langsung. Hal tersebut beresiko karena data yang dikirim bisa hilang
- b) *Unreliable*, maksudnya adalah pesan-pesan yang dikirim menggunakan protocol ini akan dikirim sebagai datagram tanpa adanya nomor urut atau pesan ACK (Acknowledgment). Hal tersebut menyebabkan pesan-pesan yang diterima di tujuan mungkin saja diterima dalam keadaan yang tidak urut sehingga protocol yang berjalan di atasnya (layer aplikasi) berperan penting dalam memulihkan pesan-pesan tersebut.

Contoh protocol aplikasi yang menggunakan UDP :

- DNS (*Domain Name System*), adalah *distribute database system* yang digunakan untuk pencarian nama computer (*Name Resolution*) di jaringan yang menggunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*).
- SNMP (*Simple Network Management Protocol*) adalah spesifikasi manajemen jaringan yang dikembangkan oleh *Internet Engineering Task Force* (IETF), sebuah bagian dari *Internet Activities Board* (IAB).
- TFTP (*Trivial File Transfer Protocol*) adalah sebuah protocol perpindahan berkas yang sangat sederhana yang memiliki fungsionalitas dasar dari protocol *File Transfer Protocol*.

3.2 PROTOKOL TCP

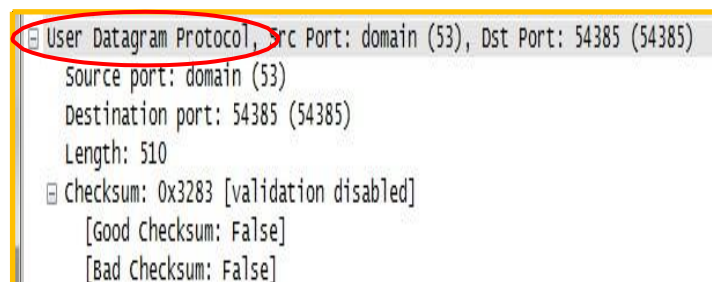
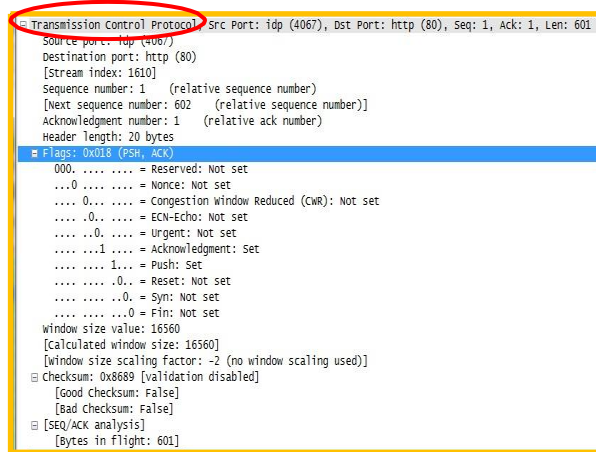
TCP merupakan protocol layer transport yang paling sering digunakan di internet. Berikut karakteristiknya :

- a) *Point to point*. Pada TCP komunikasi yang dilakukan antar user-nya bersifat point to point , maksudnya adalah satu pengiriman data dikirim oleh satu pengirim dan hanya diterima oleh satu penerima juga.
- b) *Reliable*. Berbeda dengan UDP, protocol TCP ini bersifat reliable yang artinya data yang dikirim menggunakan TCP ini akan dikirim dengan mekanisme tertentu agar data bisa diterima secara berurutan, Sehingga ada jaminan data yang dikirim dengan protocol ini bisa sampai ke tujuan dan tidak hilang.
- c) *Terdapat buffer di pengirim dan penerima*
- d) *Full duplex data*. Pada protocol ini memungkinkan adanya aliran data dua arah pada koneksi yang sama.
- e) *Connection-oriented*. Pada TCP, saat pengirim akan mengirimkan data ke penerima akan terjadi proses handshaking terlebih dahulu antara keduanya. Hal ini bertujuan agar dapat melakukan sinkronisasi terhadap nomor urut dan nomor acknowledgement yang dikirimkan kedua belah pihak dan saling bertukar ukuran TCP window.
- f) *Flow Control*. Dengan adanya flow control ini, aliran data yang dikirim akan selalu dikontrol sehingga pengirim tidak akan membuat penerima menjadi kewalahan saat menerima data-data yang dikirimkan tersebut.

Contoh Aplikasi TCP :

- HTTPs (*Hypertext Transfer Protocol Secure*) adalah ekstensi dari *Hypertext Transfer Protocol*, yang digunakan untuk komunikasi aman melalui jaringan komputer dan banyak digunakan di Internet.
- FTP merupakan seperangkat aturan yang digunakan perangkat dalam jaringan TCP/IP (Internet) untuk mentransfer file.
- SMTP Telnet (*Simple Mail Transfer Protocol*) merupakan protocol yang digunakan untuk mengirim pesan email antar server. Telnet sendiri biasanya digunakan untuk membuat koneksi ke Transmission Control Protocol (TCP).

3.3 MEMBANDINGKAN TCP DAN UDP MENGGUNAKAN WIRESHARK



Pada gambar tersebut dapat dilihat perbandingan protocol layer transport antara kedua paket tersebut. Sebelah kiri merupakan paket yang transport layer-nya menggunakan protocol TCP, sedangkan gambar sebelah kanan menggunakan protocol UDP. Pada gambar tersebut terlihat jika paket yang menggunakan protocol TCP lebih kompleks pada layer transportnya daripada UDP. Hal tersebut terjadi Karena kedua protocol ini memiliki karakteristik yang berbeda. Dapat kita lihat pada gambar TCP memiliki banyak parameter

yang tidak dimiliki oleh UDP pada detail paketnya seperti space number, header length acknowledgement number, flags, SEQ/ACK analysis, dll. Sedangkan UDP hanya memiliki beberapa parameter seperti source port, destination port, length, dan checksum.

Pada detail paket TCP terlihat jika paket tersebut memiliki sequence number dan ACK number karena pesan yang dikirim dengan TCP memiliki urutan-urutan tertentu agar lebih akurat dalam pengirimannya. Tetapi tidak dengan paket yang dikirim dengan protokol UDP. Pada paket detail-nya dapat dilihat jika pada protokol ini tidak memiliki urutan tertentu pada pengiriman paket-paketnya. Protokol ini terus menerus mengirimkan paket ke tujuan tanpa diurutkan sehingga prosesnya akan lebih cepat apabila dibandingkan dengan proses pada protokol TCP. Baru kemudian layer di atasnya, yaitu layer aplikasi yang berperan untuk mengatur paket-paket yang dikirim tersebut.

Time	169.254.17.191	169.254.202.96	fe80::596:154:9fcc:	192.168.137.15	Comment
121.709363000		indx-dds > icslap			TCP: indx-dds > icslap [SYN] Seq=0 Win=8192 Len=0 MSS=
121.710696000		icslap > indx-dds			TCP: icslap > indx-dds [SYN, ACK] Seq=0 Ack=1 Win=8192
121.710891000		indx-dds > icslap			TCP: indx-dds > icslap [ACK] Seq=1 Ack=1 Win=17520 Len=
121.711220000		SUBSCRIBE /upnphost			HTTP: SUBSCRIBE /upnphost/udhisapi.dll?event=uuid0819e
121.714449000		HTTP/1.1 200 OK			HTTP: HTTP/1.1 200 OK
121.714828000		indx-dds > icslap			TCP: indx-dds > icslap [FIN, ACK] Seq=336 Ack=250 Win=1
121.715907000		icslap > indx-dds			TCP: icslap > indx-dds [FIN, ACK] Seq=250 Ack=337 Win=1
121.716075000		indx-dds > icslap			TCP: indx-dds > icslap [ACK] Seq=337 Ack=251 Win=17271
121.746315000		wago-io-system > ic			TCP: wago-io-system > icslap [SYN] Seq=0 Win=8192 Len=
121.746979000		icslap > wago-io-sy			TCP: icslap > wago-io-system [SYN, ACK] Seq=0 Ack=1 Win=
121.747166000		wago-io-system > ic			TCP: wago-io-system > icslap [ACK] Seq=1 Ack=1 Win=174
121.747496000		SUBSCRIBE /upnphost			HTTP: SUBSCRIBE /upnphost/udhisapi.dll?event=uuid0819e
121.749490000		HTTP/1.1 200 OK			HTTP: HTTP/1.1 200 OK
121.749764000		wago-io-system > ic			TCP: wago-io-system > icslap [FIN, ACK] Seq=352 Ack=250
121.750418000		icslap > wago-io-sy			TCP: icslap > wago-io-system [FIN, ACK] Seq=250 Ack=353
121.750568000		wago-io-system > ic			TCP: wago-io-system > icslap [ACK] Seq=353 Ack=251 Win=
316.720753000		altav-remmgt > icd			TCP: altav-remmgt > icslap [SYN] Seq=0 Win=8192 Len=0

Time	169.254.202.96	239.255.255.250	169.254.17.191	fe80::596:154:9fcc:	Comment
0.000000000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
2.994392000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
5.995870000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
32.972113000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
35.972470000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
38.988241000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
42.036190000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
45.036390000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
48.060556000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
95.975351000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
98.975897000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
101.976175000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
105.024547000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
108.024077000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
111.024120000		M-SEARCH * HTTP/1.1			SSDP: M-SEARCH * HTTP/1.1
126.653587000			Domain/Workgroup An		BROWSER: Domain/Workgroup Announcement WORKGRC
161.191976000				M-SEARCH	SSDP: M-SEARCH * HTTP/1.1

UDP memiliki karakteristik yang sederhana karena protocol ini memang dibuat untuk mengoptimalkan kecepatan pengiriman datanya. Apabila protocol UDP diberi parameter yang banyak dan kompleks, maka akan terjadi delay yang cukup panjang dan akan melenceng dari tujuan utamanya yaitu mengoptimalkan kecepatan transfer data. Paket yang menggunakan protocol ini memerlukan kecepatan data yang optimal karena protocol ini biasanya digunakan pada aplikasi-aplikasi real time seperti streaming video, audio, game, dll. Dapat dilihat juga pada *flow graph*-nya jika pada UDP ini tidak terdapat proses *handshaking*, yaitu Protokol UDP ini tidak perlu membuat koneksi secara langsung terlebih dahulu untuk proses pengiriman datanya. Pada gambar *flow-graph* tersebut terlihat jika pada awal komunikasi, client langsung terus menerus mengirim pesan M-Search untuk meminta data ke server tanpa server mengirimkan ACK kembali.

Kemudian pada TCP terdapat banyak parameter pada paketnya karena protocol ini mengutamakan keandalannya dalam pengiriman data ke tujuan. Maksudnya adalah pada protocol ini terdapat jaminan bahwa paket yang dikirim akan sampai ke alamat tujuan tanpa ada kerusakan pada paket dan berurutan. Protocol ini memiliki suatu mekanisme sendiri agar paket tersebut dapat diterima secara utuh dan berurutan. Terjadi proses *handshaking* terlebih dahulu pada saat akan mengirimkan paket sehingga terjadi sinkronisasi terhadap nomor urut dan nomor acknowledgement pada kedua belah pihak. Pada *flow graph*-nya dapat dilihat jika terdapat proses pembangunan suatu koneksi dahulu antara client dan server. Client terlebih dahulu mengirimkan SYN ACK ke server kemudian server membalasnya dengan ACK, hal itu dilakukan untuk memastikan apakah kedua end user tersebut sudah benar-benar terhubung atau belum. Itulah mengapa TCP memiliki banyak parameter pada paket detailnya karena terjadi proses yang cukup kompleks pada pengiriman datanya berbeda dengan protocol UDP yang tidak menggunakan proses *handshaking* pada pengiriman data.

4. KESIMPULAN

Berdasarkan hasil kerja kelompok, kami dapat menyimpulkan bahwa Protocol UDP memiliki latency terendah dibandingkan dengan TCP, juga memiliki delay (jitter) terendah. Namun, dalam hal packet loss, UDP memiliki kecenderungan lebih besar untuk kehilangan packet dibandingkan dengan TCP. Hal ini berkaitan dengan besarnya pengguna queue yang digunakan oleh suatu jaringan akan memperbesar kemungkinan packet yang hilang.

5. REFERENSI

[1] <http://robby.c.staff.gunadarma.ac.id/Downloads/files/5659/tcpudp.pdf>

[2] <http://telecom.ee.itb.ac.id/~tutun/ET5044/0304/5.ppt>