Laporan Tugas

Konsep Jaringan Komputer

2

Disusun Oleh:

Mochamad Isya Nurdian Zawawi

NIM.E41241950 / Golongan E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2025

Chapter 3 Review Questions

SECTIONS 3.1–3.3

1. Suppose the network layer provides the following service. The network layer in the source host accepts a segment of maximum size 1,200 bytes and a destination host address from the transport layer. The network layer then guarantees to deliver the segment to the transport layer at the destination host. Suppose many network application processes can be running at the destination host.

   a. Design the simplest possible transport-layer protocol that will get application data to the desired process at the destination host. Assume the operating system in the destination host has assigned a 4-byte port number to each running application process.

   b. Modify your protocol so it also provides a "return address" to the destination process.

   c. In your protocols, does the transport layer need to do anything inside the core of the computer network?

   Jawaban:

   a. Protokol transport paling sederhana: tambahkan header berisi port tujuan (4 byte) pada setiap segmen sebelum dikirim ke network layer. Saat diterima, host tujuan akan memeriksa port tersebut dan menyerahkan data ke proses yang sesuai.

   b. Untuk menyediakan alamat pengirim (return address), tambahkan dua field port: port sumber dan port tujuan. Ini seperti format dasar UDP/TCP.

   c. Tidak. Transport layer tidak perlu melakukan apa pun di jaringan inti, karena fungsinya hanya pada end host (pengirim dan penerima).

2. Consider a planet where everyone belongs to a family of six, every family lives in its own house, each house has a unique address, and each person in a given house has a unique name. Suppose this planet has a mail service that delivers letters from source house to destination house. The mail service requires that (1) the letter be in an envelope, and that (2) the address of the destination house (and nothing more) be clearly written on the envelope. Suppose each family has a delegate family member who collects and distributes letters for the other family members. The letters do not necessarily provide any indication of the recipients of the letters.

   a. Using the solution to Problem R1 above as inspiration, describe a protocol that the delegates can use to deliver letters from a sending family member to a receiving family member.

   b. In your protocol, does the mail service ever have to open the envelope and examine the letter in order to provide its service?

   Jawaban:

   a. Setiap “delegasi keluarga” berperan seperti transport layer: ia membaca nama penerima di dalam surat (analog dengan port), dan menyerahkannya ke anggota keluarga yang tepat. Surat dikemas oleh pengirim dan diserahkan ke pos (network layer) yang hanya tahu alamat rumah, bukan siapa di dalamnya.

   b. Tidak. Layanan pos tidak perlu membuka surat. Sama seperti **network layer** tidak perlu melihat isi segmen transport.

3. Consider a TCP connection between Host A and Host B. Suppose that the TCP segments traveling from Host A to Host B have source port number x and destination port number y. What are the source and destination port numbers for the segments traveling from Host B to Host A?

   Jawaban:

   Source port = y, Destination port = x. (Kedua arah sebuah koneksi TCP memakai pasangan port yang sama namun ditukar antara sumber dan tujuan.)

4. Describe why an application developer might choose to run an application over UDP rather than TCP.

   Jawaban:

   Kontrol aplikasi lebih besar: tidak ada koneksi, retransmisi, congestion control, atau head-of-line blocking bawaan—aplikasi dapat mengatur reliability, timing, FEC, dan retransmisi selektif sendiri. Latency rendah & jitter terkendali: tanpa handshake dan tanpa menunggu kehilangan paket lama diperbaiki. Overhead kecil: header ringan, tidak ada state koneksi di endpoint maupun server (skala lebih mudah untuk traffic sederhana seperti query DNS, telemetri). Pengiriman many-to-many/broadcast via multicast lebih mudah dibanding TCP.

5. Why is it that voice and video traffic is often sent over TCP rather than UDP in today's Internet? (Hint: The answer we are looking for has nothing to do with TCP's congestion-control mechanism.)

   Jawaban:

   Aplikasi real-time memprioritaskan keterlambatan rendah dibanding keutuhan bit-per-bit. Kehilangan kecil lebih dapat diterima daripada penundaan besar akibat retransmisi. TCP menyebabkan head-of-line blocking dan menahan data berikutnya sampai paket hilang tiba—ini memperburuk latency/jitter; UDP tidak. Dengan UDP, aplikasi media dapat menyesuaikan sendiri: pacing, kodek adaptif bitrate, interleaving/FEC, atau retransmisi selektif sangat terbatas—memberi pengalaman lebih mulus. Start-up lebih cepat (tanpa 3-way handshake) dan overhead lebih kecil, bermanfaat untuk paket kecil & periodik (audio frames).

6. Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

   Jawaban:

   Ya, bisa. Walaupun UDP tidak menjamin reliabilitas, aplikasi dapat membangun reliability sendiri di layer atas. Caranya:

1) Tambahkan sequence number pada setiap paket.
2) Gunakan ACK/NAK sederhana: penerima memberi tahu paket mana yang diterima atau hilang.
3) Terapkan retransmisi jika timeout atau tidak ada ACK.
4) Bisa juga ditambah Forward Error Correction (FEC) untuk recovery sebagian error tanpa retransmisi.

Contoh nyata: QUIC (Google), SCTP user-space library, atau aplikasi game online yang mengelola reliabilitas sendiri.

7. Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B each send a UDP segment to Host C with destination port number 6789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

   Jawaban :

   Ya, keduanya akan masuk ke socket yang sama (karena socket diidentifikasi oleh port tujuan di Host C). Namun, setiap UDP segment juga memiliki source IP address dan source port pada header. Proses di Host C bisa memeriksa tuple (source IP, source port) untuk membedakan asal paket tersebut.

8. Suppose that a Web server runs in Host C on port 80. Suppose this Web server uses persistent connections, and is currently receiving requests from two different Hosts, A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain.

   Jawaban:

   Tidak, tiap koneksi TCP dibuatkan socket tersendiri. Meskipun semuanya menggunakan port 80 di sisi server, kombinasi (IP sumber, port sumber, IP tujuan, port tujuan) unik untuk tiap koneksi. Jadi Host A dan Host B masing-masing punya socket TCP-nya sendiri di server, tapi keduanya tetap “mendengarkan” di port 80.

SECTION 3.4

9. In our rdt protocols, why did we need to introduce sequence numbers?

   Jawaban:

   Karena sequence number memungkinkan penerima mengenali apakah paket yang diterima adalah duplikat atau paket baru. Tanpa sequence number, jika ACK hilang dan pengirim retransmit, penerima tidak bisa membedakan apakah itu paket lama atau baru. Sequence number juga digunakan untuk menjaga urutan pengiriman. Jadi, sequence number adalah kunci untuk deteksi duplikasi dan pengaturan urutan.

10. In our rdt protocols, why did we need to introduce timers?

    Jawaban:

    Timer diperlukan untuk mengatasi kehilangan paket atau ACK. Jika pengirim tidak menerima ACK dalam periode tertentu, pengirim akan timeout dan mengirim ulang paket. Tanpa timer, jika sebuah paket hilang di jaringan, pengirim akan menunggu selamanya. Timer memastikan retransmisi proaktif sehingga komunikasi tetap berjalan meski ada loss.

11. Suppose that the roundtrip delay between sender and receiver is constant and known to the sender. Would a timer still be necessary in protocol rdt 3.0, assuming that packets can be lost? Explain.

    Jawaban:

    Ya, timer tetap diperlukan. Walaupun pengirim tahu persis RTT, tetap ada kemungkinan paket hilang. Jika paket hilang, ACK tidak akan pernah kembali. Maka timer digunakan untuk mendeteksi kondisi ini, sehingga pengirim bisa retransmit. Jadi, pengetahuan RTT saja tidak cukup untuk menjamin reliabilitas.

12. Visit the Go-Back-N Java applet at the companion Web site.

    a. Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation. Describe what happens.

b. Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens.

c. Finally, try sending six packets. What happens?

Jawaban:

a. Jika paket pertama hilang, maka penerima hanya menerima 2–5 tapi tidak bisa meng-ACK karena Go-Back-N butuh urutan penuh. Akhirnya pengirim timeout dan mengirim ulang semua paket dari #1.

b. Jika ACK pertama hilang, pengirim timeout dan retransmit paket pertama. Penerima sudah punya #1, tapi akan terima duplikat dan drop. Proses berlanjut normal setelahnya.

c. Jika jendela ukuran N=5, maka pengirim tidak bisa mengirim 6 sekaligus. Paket ke-6 akan ditahan sampai ada ACK untuk membebaskan slot jendela.

13. Repeat R12, but now with the Selective Repeat Java applet. How are Selective Repeat and Go-Back-N different?

Jawaban:

Pada Selective Repeat, penerima bisa menyimpan paket yang datang meski ada lubang di urutan. Jadi jika paket pertama hilang, paket 2–5 tetap disimpan; hanya #1 yang perlu retransmit. Setelah #1 diterima, penerima bisa langsung menyerahkan ke aplikasi tanpa retransmit massal. Jika ACK hilang, hanya paket yang bersangkutan yang di-retransmit. Perbedaan utama: Go-Back-N retransmit semua dari titik error ke depan, sedangkan Selective Repeat hanya retransmit yang hilang.

SECTION 3.5

14. True or false?

a. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.

b. The size of the TCP rwnd never changes throughout the duration of the connection.

c. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.

d. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is m, then the sequence number for the subsequent segment will necessarily be m+1.

e. The TCP segment has a field in its header for rwnd.

f. Suppose that the last SampleRTT in a TCP connection is equal to 1 sec. The current value of TimeoutInterval for the connection will necessarily be ≥1 sec.

g. Suppose Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgment number is necessarily 42.

Jawaban:

a. False. TCP tetap mengirim ACK tanpa data (ACK standalone). Piggyback itu hanya optimisasi; kalau tak ada data, ACK tetap dikirim.

b. False. rwnd (receive window) berubah-ubah sesuai ruang buffer penerima yang tersisa.

c. True. Data "in flight" dibatasi oleh rwnd (flow control), jadi tidak boleh melebihi kapasitas yang diberitahukan penerima.

d. False. Nomor urut TCP menghitung byte, bukan paket. Jika segmen berukuran L byte, segmen berikutnya akan mulai di m + L, bukan selalu m + 1.

e. True. Di header TCP ada Window Size field yang mengiklankan receive window (rwnd) milik penerima (dengan opsi window scaling bila perlu).

f. False. TimeoutInterval = EstimatedRTT + 4 × DevRTT. EstimatedRTT merupakan smoothed RTT dan bisa lebih kecil dari SampleRTT; jika DevRTT kecil, jumlahnya bisa < 1 s. Jadi tidak "harus" ≥ 1 s.

g. False. Acknowledgment number merefleksikan next byte yang diharapkan dari lawan (Host B), bukan kelanjutan dari sequence number A. Nilainya bergantung pada data dari B yang sudah A terima, sehingga tidak harus 42.

15. Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.

    a. How much data is in the first segment?

    b. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

    Jawaban:

    a. Perbedaan sequence number = 110 – 90 = 20 byte. Jadi data dalam segmen pertama adalah 20 byte.

    b. Jika segmen pertama hilang, Host B hanya menerima segmen dengan nomor urut 110–..., tapi ia masih menunggu byte mulai dari 90. Maka ACK = 90 (ACK kumulatif selalu menunjuk byte berikutnya yang diharapkan).

16. Consider the Telnet example discussed in Section 3.5. A few seconds after the user types the letter ‘C’, the user types the letter ‘R.’ After typing the letter ‘R’, how many segments are sent, and what is put in the sequence number and acknowledgment fields of the segments?

    Jawaban:

    Saat user mengetik ‘C’: Telnet kirim 1 segmen data (berisi 1 byte, misalnya seq = 42). Penerima balas ACK = 43. Saat user mengetik ‘R’: lagi-lagi 1 segmen data (1 byte, seq = 43). Penerima balas ACK = 44. Jadi setiap karakter diketik → 2 segmen (1 data + 1 ACK). Sequence number naik +1 tiap karakter. ACK number selalu byte selanjutnya yang diharapkan.

SECTION 3.7

17. Suppose two TCP connections are present over some bottleneck link of rate R bps. Both connections have a huge file to send (in the same direction over the bottleneck link). The transmissions of the files start at the same time. What transmission rate would TCP like to give to each of the connections?

    Jawaban:

    TCP secara alami berbagi bandwidth secara fairness. Dengan 2 koneksi besar dimulai bersamaan, TCP akan berusaha agar tiap koneksi mendapat ≈ R/2 bps (asalkan RTT mirip dan loss rate serupa).

18. True or False: Consider congestion control in TCP. When the timer expires at the sender, the value of ssthresh is set to one half of its previous value.

    Jawaban:

    True. Jika terjadi timeout, TCP menganggap itu indikasi congestion berat.
    Maka: ssthresh = cwnd / 2 (dibagi dua).
    cwnd = 1 MSS (reset slow start, jika Reno/Tahoe).

19. In the discussion of TCP splitting in the sidebar in Section 3.7, it was claimed that the response time with TCP splitting is approximately 4·RTTFE+RTTBE+processing time. Justify this claim.

    Jawaban:

    Klaim bahwa waktu respons dengan TCP splitting ≈ 4·RTT_FE + RTT_BE + waktu proses bisa dijelaskan begini:

    a. RTT_FE (Front-End) = waktu bolak-balik antara klien dan proxy.
    b. RTT_BE (Back-End) = waktu bolak-balik antara proxy dan server sebenarnya.
    c. Pada proses TCP splitting, koneksi dibagi dua:
        1) TCP antara *client–proxy* (front-end).
        2) TCP antara *proxy–server* (back-end).
    d. Ada beberapa langkah handshake dan transfer di kedua sisi, sehingga total waktu tumpukannya menjadi empat kali RTT_FE (dua kali untuk

setup dan dua kali untuk transfer) ditambah satu RTT_BE untuk komunikasi proxy–server, dan ditambah processing time di proxy.

Jadi persamaan itu merepresentasikan total waktu respon dari perspektif klien yang menggunakan koneksi split TCP.

CHAPTER 4 Review Questions

SECTION 4.1

1. Let's review some of the terminology used in this textbook. Recall that the name of a transport-layer packet is segment and that the name of a link-layer packet is frame. What is the name of a network-layer packet? Recall that both routers and link-layer switches are called packet switches. What is the fundamental difference between a router and link-layer switch?

   Jawaban:

   Nama paket di network layer disebut datagram. Perbedaan mendasar antara router dan link-layer switch adalah:

   a. Router bekerja di layer 3 (network layer), memeriksa alamat IP dan menentukan rute antar jaringan.

   b. Switch bekerja di layer 2 (data-link layer), hanya memeriksa alamat MAC dan meneruskan frame dalam jaringan lokal (LAN).

   Singkatnya: router = lintas jaringan, switch = di dalam satu jaringan.

2. We noted that network layer functionality can be broadly divided into data plane functionality and control plane functionality. What are the main functions of the data plane? Of the control plane?

   Jawaban:

   Fungsi dibagi dua:

   a. Data plane: menangani pengiriman aktual paket (forwarding) dari input port ke output port — proses cepat yang berjalan di hardware.

   b. Control plane: menentukan *ke mana* paket harus pergi (routing) — dikelola oleh software/routing protocol seperti OSPF, BGP, dsb. Jadi data plane = aksi, control plane = otak pengendali.

3. We made a distinction between the forwarding function and the routing function performed in the network layer. What are the key differences between routing and forwarding?

Jawaban:

a. Forwarding: proses langsung memindahkan paket dari input ke output port berdasarkan tabel forwarding.

b. Routing: proses menentukan jalur terbaik dari sumber ke tujuan (membangun tabel forwarding).

Jadi singkatnya: Forwarding = aktivitas per-paket, Routing = aktivitas per-jaringan.

4. What is the role of the forwarding table within a router?

Jawaban:

Forwarding table adalah tabel di router yang memetakan alamat tujuan (prefix IP) ke output port. Ketika paket datang, router mencocokkan alamat IP tujuannya ke tabel ini untuk menentukan port keluar mana yang digunakan. Tabel ini biasanya dihasilkan oleh *routing protocol*.

5. We said that a network layer's service model "defines the characteristics of end-to-end transport of packets between sending and receiving hosts." What is the service model of the Internet's network layer? What guarantees are made by the Internet's service model regarding the host-to-host delivery of datagrams?

Jawaban:

Model layanan layer jaringan Internet disebut "best-effort delivery". Artinya, jaringan tidak memberikan jaminan terhadap:

a. keterurutan paket,

b. waktu pengiriman,

c. atau reliabilitas (paket bisa hilang).

Yang dijamin hanya: paket akan diusahakan dikirim sebaik mungkin dari host sumber ke host tujuan — tanpa kepastian pasti sampai.

SECTION 4.2

6. In Section 4.2, we saw that a router typically consists of input ports, output ports, a switching fabric and a routing processor. Which of these are implemented in hardware and which are implemented in software? Why?

Returning to the notion of the network layer's data plane and control plane, which are implemented in hardware and which are implemented in software? Why?

Jawaban:

a. Input port, output port, dan switching fabric umumnya diimplementasikan dengan hardware (ASIC/FPGA) karena harus bekerja sangat cepat memproses paket dalam nanodetik.

b. Routing processor diimplementasikan dalam software karena bertugas menghitung rute dan mengatur tabel forwarding, bukan menangani paket langsung.

   Jadi:

   Data plane → hardware (karena performa tinggi).

   Control plane → software (karena butuh fleksibilitas dan logika routing).

7. Discuss why each input port in a high-speed router stores a shadow copy of the forwarding table.

   Jawaban:

   Setiap input port menyimpan salinan bayangan (shadow copy) dari *forwarding table* supaya tidak perlu bolak-balik ke *routing processor*. Tujuannya agar lookup alamat tujuan bisa dilakukan langsung di port dengan kecepatan tinggi, tanpa menunggu CPU utama.

8. What is meant by destination-based forwarding? How does this differ from generalized forwarding (assuming you've read Section 4.4, which of the two approaches are adopted by Software-Defined Networking)?

   Jawaban:

   a. Destination-based forwarding: router memilih output port hanya berdasarkan alamat tujuan IP.

   b. Generalized forwarding: router bisa membuat keputusan berdasarkan banyak field (bukan hanya IP), misalnya jenis protokol, port TCP/UDP, atau label tertentu.

Software-Defined Networking (SDN) menggunakan generalized forwarding.

9. Suppose that an arriving packet matches two or more entries in a router's forwarding table. With traditional destination-based forwarding, what rule does a router apply to determine which of these rules should be applied to determine the output port to which the arriving packet should be switched?

   Jawaban:

   Kalau paket cocok dengan lebih dari satu entri di *forwarding table*, router tradisional memakai aturan Longest Prefix Match (LPM) — yaitu memilih entri dengan prefix IP terpanjang (paling spesifik). Ini memastikan rute yang paling tepat dipilih.

10. Three types of switching fabrics are discussed in Section 4.2. List and briefly describe each type. Which, if any, can send multiple packets across the fabric in parallel?

    Jawaban:

    Tiga jenis switching fabric:

    1) Memory-based switching: paket disimpan di memori sentral; dilakukan oleh CPU. (Lambat, model lama)
    2) Bus-based switching: paket dikirim melalui bus bersama antar port. (Lebih cepat, tapi terbatas bandwidth)
    3) Interconnection network (crossbar): koneksi langsung antar input–output, bisa melakukan multiple transfer paralel. (Paling cepat dan efisien)

    Yang bisa kirim banyak paket paralel = interconnection network.

11. Describe how packet loss can occur at input ports. Describe how packet loss at input ports can be eliminated (without using infinite buffers).

    Jawaban:

    Packet loss di input port terjadi ketika paket menunggu untuk masuk ke fabric, tapi buffer penuh.

Cara menghindarinya: gunakan scheduling dan buffer management yang efisien seperti *input queuing dengan prioritas* atau *virtual output queuing (VOQ)* agar antrean tidak menumpuk di satu port.

12. Describe how packet loss can occur at output ports. Can this loss be prevented by increasing the switch fabric speed?

    Jawaban:

    Packet loss di output port terjadi saat banyak input port mengirim ke output port yang sama dan buffer output penuh. Meningkatkan kecepatan fabric tidak selalu mencegah loss, karena masalahnya ada di batas buffer output, bukan di kecepatan switch.

13. What is HOL blocking? Does it occur in input ports or output ports?

    Jawaban:

    HOL (Head-of-Line) blocking terjadi ketika paket pertama dalam antrean menahan paket di belakangnya meskipun yang di belakang sebenarnya bisa diteruskan ke output lain. Ini terjadi di input port, bukan di output port.

14. In Section 4.2, we studied FIFO, Priority, Round Robin (RR), and Weighted Fair Queueing (WFQ) packet scheduling disciplines. Which of these queueing disciplines ensure that all packets depart in the order in which they arrived?

    Jawaban:

    Disiplin antrian yang menjamin urutan paket keluar sama seperti saat masuk adalah FIFO (First In, First Out). Sementara Priority, RR, dan WFQ bisa mengubah urutan pengiriman karena mempertimbangkan prioritas atau bobot.

15. Give an example showing why a network operator might want one class of packets to be given priority over another class of packets.

    Jawaban:

    Contoh: operator jaringan ingin paket video conference dikirim lebih cepat daripada email. Maka mereka memberi prioritas lebih tinggi ke paket real-time agar tidak delay, sementara paket lain bisa menunggu.

16. What is an essential different between RR and WFQ packet scheduling? Is there a case (Hint: Consider the WFQ weights) where RR and WFQ will behave exactly the same?

    Jawaban:

    Perbedaan utama:

    a. Round Robin (RR): setiap antrean diberi giliran pengiriman secara merata.
    b. Weighted Fair Queueing (WFQ): giliran disesuaikan dengan bobot (ada antrean yang mendapat jatah lebih besar).

    RR dan WFQ akan berperilaku sama jika semua bobot di WFQ sama besar (misalnya semua bobot = 1).

SECTION 4.3

17. Suppose Host A sends Host B a TCP segment encapsulated in an IP datagram. When Host B receives the datagram, how does the network layer in Host B know it should pass the segment (that is, the payload of the datagram) to TCP rather than to UDP or to some other upper-layer protocol?

    Jawaban:

    Ketika Host B menerima IP datagram, network layer melihat field "Protocol" di header IP untuk menentukan apakah payload harus dikirim ke TCP (value = 6), UDP (value = 17), atau protokol lain. Jadi field *Protocol* yang memberi tahu paket itu milik layer transport mana.

18. What field in the IP header can be used to ensure that a packet is forwarded through no more than N routers?

    Jawaban:

    Field di header IP yang membatasi jumlah router adalah Time To Live (TTL). Nilai TTL dikurangi setiap kali datagram melewati router. Jika TTL mencapai 0, paket dibuang — mencegah paket berputar tanpa henti.

19. Recall that we saw the Internet checksum being used in both transport-layer segment (in UDP and TCP headers, Figures 3.7 and 3.29

respectively) and in network-layer datagrams (IP header, Figure 4.16). Now consider a transport layer segment encapsulated in an IP datagram. Are the checksums in the segment header and datagram header computed over any common bytes in the IP datagram? Explain your answer.

Jawaban:

Checksum di header IP dan checksum di header TCP/UDP tidak dihitung dari byte yang sama. Checksum IP hanya untuk header IP, sementara checksum TCP/UDP dihitung atas header + data + pseudo-header (yang berisi IP sumber dan tujuan). Jadi mereka tidak tumpang tindih, tapi saling melengkapi.

20. When a large datagram is fragmented into multiple smaller datagrams, where are these smaller datagrams reassembled into a single larger datagram?

    Jawaban:

    Ketika datagram besar dipecah jadi beberapa fragment, reassembly (penggabungan ulang) dilakukan di host tujuan, bukan di router tengah. Router hanya memecah, tidak menggabung kembali.

21. Do routers have IP addresses? If so, how many?

    Jawaban:

    Ya, router memiliki IP address pada setiap interface-nya. Jadi kalau router punya 4 interface, dia punya 4 IP address — satu per jaringan yang dihubungkan.

22. What is the 32-bit binary equivalent of the IP address 223.1.3.27?

    Jawaban:

    11011111.00000001.00000011.00011011

23. Visit a host that uses DHCP to obtain its IP address, network mask, default router, and IP address of its local DNS server. List these values.

    Jawaban:

    Nilai yang diminta (IP, subnet mask, default gateway, DNS) tergantung jaringan lokal masing-masing. Contoh hasil DHCP biasa:

    a. IP Address: 192.168.1.10

b. Subnet Mask: 255.255.255.0

c. Default Router: 192.168.1.1

d. DNS Server: 8.8.8.8

(Ini hanya contoh umum dari DHCP client.)

24. Suppose there are three routers between a source host and a destination host. Ignoring fragmentation, an IP datagram sent from the source host to the destination host will travel over how many interfaces? How many forwarding tables will be indexed to move the datagram from the source to the destination?

Jawaban:

Dengan 3 router: jalur = Source → R1 → R2 → R3 → Destination.

Interfaces dilewati:

Source NIC (1), R1 in + out (2), R2 in + out (2), R3 in + out (2) , Destination NIC (1). Total = 8 interfaces. Forwarding tables yang diakses: 3 (satu di setiap router: R1, R2, R3).

25. Suppose an application generates chunks of 40 bytes of data every 20 msec, and each chunk gets encapsulated in a TCP segment and then an IP datagram. What percentage of each datagram will be overhead, and what percentage will be application data?

Jawaban:

Header TCP = 20 byte, header IP = 20 byte → total header = 40 byte.

Data aplikasi = 40 byte. Jadi:

a. Overhead = 40 / (40+40) × 100% = 50%

b. Data aplikasi = 50%

26. Suppose you purchase a wireless router and connect it to your cable modem. Also suppose that your ISP dynamically assigns your connected device (that is, your wireless router) one IP address. Also suppose that you have five PCs at home that use 802.11 to wirelessly connect to your wireless router. How are IP addresses assigned to the five PCs? Does the wireless router use NAT? Why or why not?

Jawaban:

Router rumah menerima 1 IP publik dari ISP. Router itu kemudian memberikan IP privat (mis. 192.168.1.x) ke perangkat lokal lewat DHCP. Untuk bisa terkoneksi ke Internet, router menggunakan NAT (Network Address Translation) agar banyak perangkat bisa berbagi satu IP publik. Jadi, ya, router menggunakan NAT.

27. What is meant by the term “route aggregation”? Why is it useful for a router to perform route aggregation?

Jawaban:

Route aggregation = penggabungan beberapa prefix routing menjadi satu entri yang lebih umum (supernetting). Contoh: 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24 bisa digabung jadi 192.168.0.0/22.

Manfaat:

a. Mengurangi ukuran tabel routing.

b. Mempercepat pencarian route.

c. Skalabilitas routing Internet (mengurangi BGP entries).

28. What is meant by a “plug-and-play” or “zeroconf” protocol?

Jawaban:

Plug-and-play (zeroconf) = host bisa mengkonfigurasi dirinya otomatis tanpa administrator. Termasuk: mendapatkan IP address, gateway, DNS, tanpa harus manual. Contoh: DHCP, atau di Windows → APIPA (Automatic Private IP Addressing) 169.254.x.x.

29. What is a private network address? Should a datagram with a private network address ever be present in the larger public Internet? Explain.

Jawaban:

Private IP address ranges (RFC 1918):

IP ADDRESS	SUBNETMASK
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

Hanya digunakan dalam local networks. Datagram dengan private IP tidak boleh muncul di Internet publik, karena tidak bisa dirouting secara global. Jika ingin akses Internet, private IP harus ditranslasi ke public IP via NAT.

30. Compare and contrast the IPv4 and the IPv6 header fields. Do they have any fields in common?

    Jawaban:

    a. IPv4 header: 20–60 bytes, banyak field (version, IHL, TOS, length, ID, flags, fragment offset, TTL, protocol, checksum, src IP, dst IP, options).

    b. IPv6 header: 40 bytes fixed, lebih sederhana (version, traffic class, flow label, payload length, next header, hop limit, src IP, dst IP).

    c. Field yang sama: Version, Src Address, Dst Address, Traffic Class (≈ TOS), Hop Limit (≈ TTL), Next Header (≈ Protocol).

    IPv6 membuang field yang dianggap tidak efisien (misalnya checksum, fragment di header utama).

31. It has been said that when IPv6 tunnels through IPv4 routers, IPv6 treats the IPv4 tunnels as link-layer protocols. Do you agree with this statement? Why or why not?

    Jawaban:

    Ya, benar. Saat IPv6 dikirim melalui IPv4 tunnel, paket IPv6 dienkapsulasi di dalam payload IPv4. Dari sudut pandang IPv6, IPv4 tunnel hanyalah link-layer virtual yang menghubungkan dua node IPv6.

    Jadi tunneling = IPv6 memperlakukan IPv4 sebagai media/link.

SECTION 4.4

32. How does generalized forwarding differ from destination-based forwarding?

    Jawaban:

    Destination-based forwarding: forwarding hanya berdasarkan alamat tujuan (IP dest). Generalized forwarding: forwarding bisa berdasarkan berbagai header field (src/dst IP, port, protokol, DSCP, dll). Generalized forwarding = lebih fleksibel, digunakan di SDN (Software-Defined Networking).

33. What is the difference between a forwarding table that we encountered in destination-based forwarding in Section 4.1 and OpenFlow's flow table that we encountered in Section 4.4?

    Jawaban:

    Forwarding table (tradisional): berisi mapping prefix alamat → output interface. OpenFlow flow table: berisi aturan lebih kompleks → mencocokkan berbagai field (IP, MAC, port, protocol) dan menentukan action (forward, drop, modify). Flow table = generalisasi forwarding table.

34. What is meant by the "match plus action" operation of a router or switch? In the case of destination-based forwarding packet switch, what is matched and what is the action taken? In the case of an SDN, name three fields that can be matched, and three actions that can be taken.

    Jawaban:

    Match plus action: inti generalized forwarding → perangkat mencocokkan field tertentu di header, lalu melakukan aksi tertentu. Tradisional (dest-based): match = destination IP prefix, action = kirim ke output interface. SDN (OpenFlow): contoh Match fields: src IP, dst IP, TCP port, MAC address, VLAN ID, protocol. Actions: forward ke port tertentu, drop paket, rewrite header, mirror, encapsulate.

35. Name three header fields in an IP datagram that can be "matched" in OpenFlow 1.0 generalized forwarding. What are three IP datagram header fields that cannot be "matched" in OpenFlow?

    Jawaban:

    Dapat dicocokkan (matched):

    a. Source IP address
    b. Destination IP address
    c. Protocol field

    Tidak dapat dicocokkan:

    a. Header checksum

b. Fragment offset
c. Flags (misalnya MF/DF)

OpenFlow fokus pada field yang relevan untuk forwarding, bukan field yang bersifat kontrol internal.