PRAKTIKUM
DESAIN DAN MANAJEMEN JARINGAN KOMPUTER

Nama	**Aliyah Rizky Al-Afifah Polanda**	**No. Modul**	**06**
NPM	**2206024682**	**Tipe**	**Tugas Pendahuluan**

1. *Quality of Service* (QoS) merupakan mekanisme penting yang digunakan untuk menjamin kualitas dan kinerja dari sebuah jaringan. Hal ini berkaitan dengan keberhasilan pengiriman paket, termasuk pengiriman melalui jaringan dengan kapasitas yang terbatas. Menerapkan QoS dapat berguna untuk meningkatkan kinerja aplikasi dan mengelola lalu lintas dengan lebih baik, menjamin kualitas layanan yang diinginkan, serta memberikan pengalaman pengguna yang memuaskan. Dibawah ini adalah beberapa kasus yang menunjukkan kebutuhan akan QoS:
   a. Video dan konferensi audio memerlukan keterlambatan dan tingkat kehilangan paket (*packet loss*) yang terbatas.
   b. *Audio* dan *video streaming* membutuhkan tingkat kehilangan paket yang terbatas.
   c. Aplikasi *real-time* sensitif terhadap keterlambatan.
   d. *Valuable applications* harus diberikan layanan yang lebih baik daripada aplikasi lain.

   Dengan menerapkan QoS, jaringan dapat memberikan prioritas kepada paket data yang kritis seperti aplikasi *real-time* atau layanan VoIP, sehingga pengiriman dapat dilakukan dengan tepat waktu dan tanpa gangguan. Selain itu, pengaturan batas penggunaan *bandwidth* dapat diterapkan untuk setiap jenis aplikasi, sehingga kemungkinan kelebihan beban dalam transmisi data dapat diminimalkan. Penggunaan QoS menjadi kunci dalam optimalisasi penggunaan sumber daya jaringan dan menjaga kualitas layanan, bahkan dalam kondisi jaringan yang padat atau terbatas.

   Referensi:

   - "Computer Network | Quality of Service and Multimedia," geeksforgeeks.org, Nov. 2021. [Online]. Available: https://www.geeksforgeeks.org/computer-network-quality-of-service-and-multimedia/. [Accessed Apr. 04, 2024].
   - "What Is Quality of Service (QoS) in Networking?" fortinet.com. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/qos-quality-of-service. [Accessed Apr. 04, 2024].

2. *Traffic* dalam jaringan mengacu pada aliran data yang dipertukarkan antara perangkat-perangkat dalam

jaringan tersebut. Setiap *traffic* memiliki prioritas yang berbeda-beda. Terdapat tiga tipe *traffic* yang umum digunakan, diurutkan dari yang memiliki prioritas paling tinggi:

a. **Voice traffic**.

Meliputi aplikasi komunikasi suara seperti VoIP (*Voice over Internet Protocol*). *Voice traffic* bersifat *real-time* dan termasuk tipe yang sensitif, sehingga paket harus dikirimkan dalam waktu yang bersamaan atau dengan *delay* yang sangat kecil. Selain itu, permasalahan yang juga perlu diperhatikan adalah tingkat kehilangan paket. Yang dapat ditoleransi adalah kehilangan paket dalam jumlah yang kecil, Jika kehilangan paket terjadi pada tingkat yang tinggi, maka komunikasi yang terjadi menjadi tidak diapahami.

Voice traffic memiliki prioritas yang paling tinggi. Hal ini karena diperlukan pengiriman paket yang tepat waktu untuk menjaga kualitas suara yang jernih. Keterlambatan dan kehilangan paket dapat menyebabkan percakapan terpotong, sehingga mengganggu komunikasi yang berlangsung.

b. **Video traffic**.

Meliputi *video streaming* dan *video conference*. Merupakan *traffic* dengan volume tinggi, namun tidak terlalu sensitif seperti *voice traffic*, karena umumnya *video traffic* tidak bersifat *real-time.* Toleransi keterlambatan paket dapat dilakukan hingga tingkat tertentu, karena tidak menimbulkan gangguan yang signifikan, hanya menyebabkan adanya waktu tambahan pengiriman paket. Meskipun terjadi kehilangan paket dalam jumlah tertentu, video masih dapat dimengerti dengan jelas. Untuk *video conference*, merupakan *traffic* yang bersifat *real-time* sehingga keterlambatan menjadi lebih penting.

Video traffic memiliki tingkat prioritas setelah *voice traffic*. Hal ini karena video membutuhkan pengiriman paket yang konsisten untuk menjaga kualitas tampilan yang baik. Keterlambatan tidak menyebabkan dampak seperti pada *voice traffic*, melainkan hanya berdampak pada munculnya *stuttering* atau pengurangan kualitas video yang ditampilkan.

c. Data traffic.

Merupakan *traffic* yang tidak sensitif terhadap kehilangan paket data karena dapat dilakukan pengiriman ulang jika hal tersebut terjadi. *Data traffic* digunakan dalam *email*, *file transfer*, dan *webpage*. Protokol yang digunakan adalah TCP, pengiriman ulang paket dapat bermanfaat untuk mengurangi kerugian dari kehilangan paket. Keterlambatan juga tidak terlalu berdampak pada pengiriman paket data. *Data traffic* memiliki tingkat prioritas yang paling rendah karena tidak sensitif terhadap permasalahan keterlambatan dan kehilangan paket data.

Referensi:

- “Network Traffic Types,” ipcisco.com. [Online]. Available: https://ipcisco.com/lesson/network-

traffic-types/. [Accessed Apr. 04, 2024].

3. Untuk menyediakan QoS yang baik, terdapat beberapa faktor yang perlu diperhatikan, yaitu *bandwidth*, *delay*, *jitter*, dan *packet loss*. Deskripsi dari masing-masing faktor tersebut adalah:
   a. *Bandwidth*: kapasitas maksimum dari suatu jaringan untuk melakukan transmisi data.
   b. *Delay*/*latency*: waktu yang dibutuhkan dalam perjalanan data dari satu titik ke titik tujuan.
   c. *Jitter*: variasi waktu tiba paket antara satu paket dengan paket lainnya.
   d. *Packet loss*: tingkat kehilangan paket data selama pengiriman terjadi.

   Dengan deskripsi tersebut, dapat disimpulkan bahwa *bandwidth* menunjukkan seberapa banyak data dapat dilewati dalam jaringan, *delay* menunjukkan total waktu pengiriman data, *jitter* menunjukkan ketidakstabilan dalam waktu kedatangan paket data, sedangkan *packet los*s menunjukkan persentase hilangnya paket data selama proses transmisi berlangsung.

   Referensi:

   - "Difference between Latency and Jitter," javatpoint.com. [Online]. Available: https://www.javatpoint.com/latency-vs-jitter. [Accessed Apr. 04, 2024].
   - "How to Describe Network Performance?" baeldung.com, Mar. 2024. [Online]. Available: https://www.baeldung.com/cs/bandwidth-packet-loss-latency-jitter. [Accessed Apr. 04, 2024].

4. *Network congestion* merupakan permasalahan yang terjadi ketika *traffic* yang mengalir dalam jaringan melebihi kapasitas maksimum dari jaringan tersebut. Permasalan ini dapat menyebabkan menumpuknya paket data pada satu atau beberapa titik sehingga terjadi keterlambatan atau kehilangan dalam pengiriman paket data. Beberapa penyebab dari terjadinya *network congestion* adalah sebagai berikut:
   a. Konsumsi *bandwidth* yang berlebihan.

      Terjadi saat terdapat penggunaan *bandwidth* yang melebihi kapasitas oleh seorang pengguna atau suatu perangkat dalam jaringan. Contoh aktivitas yang dapat menyebabkan permasalahan tersebut adalah *video streaming* atau transfer file berukuran besar. Hal ini dapat berakibat pada pemberian beban berlebih pada peralatan jaringan seperti *router* atau *switch*, sehingga berakhir pada munculnya *network congestion*.
   b. Manajemen *subnet* yang buruk.

      Maksudnya adalah penerapan *subnet* yang tidak sesuai dengan pola penggunaan dan kebutuhan akan sumber daya. Dapat menyebabkan *bottlenecks* atau titik-titik kemacetan dalam jaringan. Hal ini dihasilkan dari ketidakseimbangan dalam aliran lalu lintas *subnet* yang berkaitan.

c. *Broadcast storm.*

Terjadi saat munculnya peningkatan permintaan dalam jaringa. Selain itu dapat terjadi karena adanya *loop* dalam jaringan yang menghasilkan pesan yang sama dikirimkan berulang kali. Akibatnya. Pesan-pesan tersebut akan memenuhi jaringan dan *network congestion* akan terjadi.

d. Perangkat keras yang kadaluwarsa.

Perangkat jaringan yang telah lama digunakan dapat memiliki penurunan efisiensi. Jika digunakan untuk mengirimkan paket berukuran besar, maka dapat berpotensi untuk menyebabkan kemacetan dalam jaringan. Hal ini karena transmisi data dapat terhambat dan melambat akibat penurunan efisiensi perangkat tersebut.

e. Banyaknya perangkat dalam jaringan.

Setiap jaringan memiliki batasan akan kapasitas data yang dapat ditangani. Semakin banyak perangkat yang ada dalam suatu jaringan, maka semakin banyak *traffic* yang akan dipertukarkan antar perangkat-perangkat tersebut. Jika tidak dilakukan pengelolaan jaringan yang baik, maka jenis jaringan ini memiliki risiko yang tinggi dalam menyebabkan *network congestion*.

Referensi:

- "What is Network Congestion? Common Causes and How to Fix Them?" geeksforgeeks.org, Feb. 2023. [Online]. Available: https://www.geeksforgeeks.org/what-is-network-congestion-common-causes-and-how-to-fix-them/. [Accessed Apr. 14, 2024].

5. Jenis-jenis *delay*:

a. **Code (processing) delay**.

Merupakan waktu yang diperlukan untuk memproses paket, seperti waktu yang diperlukan oleh *router* untuk menentukan tujuan selanjutnya (*next hop*) dari paket tersebut. Selain itu *code delay* juga meliputi pembaruan TTL dan kalkulasi *checksum* oleh *router*. Besarnya nilai *code delay* bergantung pada kecepatan dari prosesor yang melakukan pemrosesan paket data.

b. Packetization delay.

Paket data tidak langsung dikirimkan dalam ukuran yang besar, melainkan dapat dibagi menjadi beberapa paket kecil. Waktu yang dibutuhkan untuk melakukan proses tersebut merupakan *packetization delay*. Pembagian paket menjadi paket-paket yang lebih kecil berguna untuk menyeimbangkan ukuran paket dengan CPU *load.*

c. Queuing delay.

Paket yang telah sampai di tujuan tidak langsung di proses oleh penerima. Terdapat antrian, dimana paket yang tiba diawal akan diproses pertama. Paket-paket ini berada dalam antrian yang disebut

sebagai *buffer*. Waktu yang diperlukan paket untuk menunggu dalam buffer tersebut, hingga akhirnya akan diproses oleh penerima, disebut sebagai *queuing delay*. Semakin banyak paket yang berada dalam antrian, maka nilai *queuing delay* akan semakin besar. Paket yang tiba tanpa adanya interval waktu atau penerima yang memiliki *server*/*links* dalam jumlah terbatas, juga dapat menyebabkan peningkatan *queuing delay.*

d. Serialization delay.

Merupakan waktu yang dibutuhkan untuk melakukan konversi data dari bentuk paralel menjadi bentuk serial, sebelum dikirimkan melalui media transmisi tertentu. Juga dapat dikatakan bahwa *serialization delay* adalah waktu yang diperlukan untuk memasukkan *voice/data frame* ke *network interface*. Jenis *delay* ini berhubungan langsung dengan *clock rate* di *trunk*.

e. Propagation delay.

Merupakan *delay* yang terjadi karena adanya waktu yang diperlukan oleh paket untuk bergerak dari sumber ke tujuan melalui media tertentu. Setelah paket dikirimkan melalui media transmisi, akan muncul *propagation delay*, karena paket (hingga bit terakhir) harus sampai ke tujuan. Terdapat dua faktor yang mempengaruhi jenis ini, yaitu jarak dan kecepatan. Jika jarak antara sumber dan tujuan semakin jauh, maka waktu yang diperlukan untuk sampai ke tujuan juga bertambah. Jika kecepatan dari media transmisi yang digunakan semakin cepat, maka waktu yang diperlukan untuk sampai ke tujuan akan berkurang.

f. De-jitter delay.

Merupakan waktu yang dibutuhkan untuk mengurangi efek *jitter* pada jaringan. Hal ini dilakukan dengan mengubah *variable delay* menjadi *fixed delay* oleh sebuah *buffer* yang disebut sebagai *de-jitter buffer*. *De-jitter buffer* harus diperhatikan dengan baik, karena jika *buffer* tersebut hanya menyimpan data dalam waktu yang singkat, maka dapat muncul potensi kesenjangan dalam paket yang ditransmisikan.

Referensi:

- "Delays in Computer Network," geeksforgeeks.org, Apr. 2023. [Online]. Available: https://www.geeksforgeeks.org/delays-in-computer-network/. [Accessed Apr. 15, 2024].
- "Understanding Delay in Packet Voice Network," cisco.com, Feb. 2006. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/5125-delay-details.html. [Accessed Apr. 15, 2024].

6. *Queuing algorithm*:
   a. FIFO (*First In First Out*).

Merupakan algoritma yang paling sederhana dalam mengelola *congestion* (kemacetan) dalam jaringan. Setiap *traffic* diperlakukan secara sama, tanpa memberikan prioritas pada jenis *traffic* tertentu. Paket-paket berada dalam suatu antrian dan akan diproses berdasarkan urutan waktu tiba. Paket yang sampai pertama akan diproses paling awal.

b. WFQ (*Weighted Fair Queuing*).

Merupakan algoritma yang memberikan prioritas tertentu ke setiap *traffic* yang berbeda. Prioritas *traffic* ini akan menentukan alokasi *bandwidth* yang diberikan untuk mentransmisikan jenis *traffic* tersebut. Alokasi *bandwidth* tersebut berguna untuk mengurangi *response time.*

c. Class-based Weighted Queuing.

Merupakan pengembangan dari WFQ. Memberikan dukungan untuk kelas-kelas berdasarkan *traffic* yang ditentukan oleh pengguna. Hal ini memberikan kebebasan yang lebih kepada administrator jaringan untuk mengelompokkan *traffic* dan menentukan prioritasnya.

d. Low Latency Queuing.

Merupakan algoritma yang dirancang untuk aplikasi *real-time* seperti VoIP dan *video streaming.* Jenis *traffic* tersebut akan memiliki prioritas yang paling tinggi. Menyebabkan paket yang termasuk dalam jenis tersebut akan ditransmisikan paling awal.

Perbedaan setiap jenis algoritma:

FIFO (*First In First Out*)	WFQ (*Weighted Fair Queuing*)	*Class-based Weighted Queuing*	*Low Latency Queuing*
Mengirimkan paket berdasarkan urutan kedatangan.	Memberikan bobot berbeda pada setiap *traffic*.	Membagi *traffic* menjadi beberapa kelas dan pengiriman paket akan dilakukan berdasarkan prioritas kelas-kelas tersebut.	Memberikan prioritas pada aplikasi *real-time.*
Mudah untuk diimplementasikan	Lebih kompleks.	Lebih kompleks.	Lebih kompleks.
Tidak memberikan prioritas pada paket.	Memastikan porsi yang adil untuk setiap *traffic*	Memungkinkan administrator untuk menentukan prioritas dengan lebih spesifik.	Mengurangi latensi untuk aplikasi *real-time.*

Referensi:

- "Class-Based Weighted Fair Queueing," cisco.com. [Online]. Available:

https://www.cisco.com/en/US/docs/ios/12_0t/12_0t5/feature/guide/cbwfq.html. [Accessed Apr. 15, 2024].

- "QUALITY OF SERVICE, PART 9 – FIFO QUEUING," globalknowledge.com. [Online]. Available: https://www.globalknowledge.com/ca-en/resources/resource-library/articles/quality-of-service-part-9-fifo-queuing/. [Accessed Apr. 15, 2024].
- "QUALITY OF SERVICE, PART 10 – WEIGHTED FAIR QUEUING," globalknowledge.com. [Online]. Available: https://www.globalknowledge.com/ca-en/resources/resource-library/articles/quality-of-service-part-10-weighted-fair-queuing/. [Accessed Apr. 15, 2024].

7. Terdapat tiga model QoS yang umum untuk digunakan, yaitu *best effort model*, *integrated services* (intServ) *model*, dan *differentiated services* (diffServ) *model*. Ketiganya memiliki perbedaan dalam mekanisme untuk memberikan jaminan QoS yang konsisten. Pemilihan model diantara ketiganya harus disesuaikan dengan kebutuhan jaringan dan aplikasi/layanan yang berjalan diatasnya.

   Perbedaan antara best effort model, intServ model, dan diffServ model:

Best Effort Model	*IntServ Model*	*DiffServ Model*
Tidak memberikan prioritas pada aplikasi atau jenis *traffic* tertentu.	Memberikan QoS untuk aplikasi *real-time*.	Memberikan QoS dengan cara membagi *traffic* ke beberapa kelas berbeda.
Tidak membutuhkan konfigurasi tambahan.	Memiliki konfigurasi yang kompleks dan butuh pengelolaan lebih lanjut.	Memiliki konfigurasi yang lebih sederhana daripada intServ model.
Sesuai untuk jaringan yang tidak membutuhkan QoS konsisten.	Kurang sesuai untuk jaringan dengan skala besar.	Sesuai untuk jaringan dengan skala besar.

   Best effort model mengirimkan data tanpa memberikan jaminan kualitas layanan. intServ model memberikan kualitas layanan untuk aplikasi *real-time* seperti *voice* dan *video traffic* dengan mengatur reservasi ke sumber daya jaringan. Sedangkan diffServ model memberikan kualitas layanan dengan membagi lalu lintas ke dalam kelas berbeda, tanpa perlu pengaturan sumber daya yang rumit.

   Referensi:

- "Diffserv QoS vs IntServ QoS Models," study-ccnp.com. [Online]. Available: https://study-ccnp.com/diffserv-qos-intserv-qos-models/. [Accessed Apr. 15, 2024].

8. QoS model:
   a. *Best-effort model*.

      Kelebihan:
      - Mudah untuk diimplementasikan karena sederhana dan tidak memerlukan alokasi sumber daya yang kompleks.
      - Sesuai diterapkan pada aplikasi yang tidak membutuhkan QoS yang konsisten.

      Kekurangan:
      - Tidak memberikan jaminan terhadap kualitas dan kinerja jaringan.
      - Tidak memberikan prioritas pada aplikasi atau layanan yang membutuhkannya.

   b. *Integrated service model*.

      Kelebihan:
      - Memberikan QoS yang lebih baik untuk aplikasi *real-time.*
      - Memungkinkan reservasi sumber daya yang tepat untuk setiap lalu lintas.

      Kekurangan:
      - Membutuhkan *overhead* tambahan dan memiliki manajemen yang kompleks untuk mengatur RSVP.
      - Tidak sesuai untuk jaringan yang berskala besar, karena membutuhkan pengelolaan yang terperinci.

   c. *Differentiated services model*.

      Kelebihan:
      - Memungkinkan diferensiasi prioritas dengan mengelompokkan lalu lintas ke dalam beberapa kelas.
      - Lebih mudah dikelola daripada *integrated service model* karena menggunakan pendekatan yang lebih sederhana.

      Kekurangan:
      - Jaminan terhadap QoS yang diberikan tidak lebih baik daripada *integrated service model*.
      - Tidak sesuai untuk digunakan pada aplikasi atau layanan yang membutuhkan tingkat QoS yang tinggi.

   Referensi:

   - "Diffserv QoS vs IntServ QoS Models," study-ccnp.com. [Online]. Available: https://study-ccnp.com/diffserv-qos-intserv-qos-models/. [Accessed Apr. 15, 2024].

9. QoS *sequence* merupakan urutan langkah yang dilakukan untuk menerapkan QoS dalam jaringan.

Terdapat tiga kategori QoS tools, yaitu:

a. **Classification and marking tools.**

   Digunakan untuk mengklasifikasikan *traffic* ke dalam kriteria tertentu. Klasifikasi ini dapat dilakukan berdasarkan alamat IP, *port*, atau jenis aplikasi/layanan. Sesi atau aliran paket akan dianalisis untuk menentukan kelas lalu lintasnya. Setelah kelas ditentukan, maka paket-paket tersebut akan ditandai (*marked*). Paket akan diperlakukan sesuai dengan tanda yang diberikan kepadanya.

b. **Congestion avoidance tools.**

   Digunakan untuk menghindari *congestion* dalam jaringan dengan memberlakukan paket-paket yang berada dalam jaringan sesuai dengan klasisfikasinya. Untuk menghindari *congestion*, paket-paket yang berada dalam jaringan dapat diberi perlakuan seperti dihilangkan (*dropped*), ditunda (*delayed*), dan ditandai ulang (*re-marked*).

c. **Congestion management tools.**

   Digunakan untuk mengatasi terjadinya *congestion* dalam jaringan. Paket-paket telah diberi prioritas berdasarkan klasifikasi yang dilakukan sebelumnya. Nilai prioritas ini akan digunakan untuk mengelola setiap paket, sehingga *congestion* dapat diatasi. Teknik yang biasa diterapkan untuk mengatasi *congestion* adalah *queuing* dan *scheduling* paket. Hal ini penting untuk memastikan bahwa paket dengan prioritas tinggi tetap dapat sampai ke tujuan.

Dari penjelasan diatas, dapat disimpulkan bahwa *classification and marking tools* digunakan untuk mengklasifikasikan dan menandari *traffic* jaringan berdasarkan kriteria tertentu. *Congestion avoidance tools* digunakan untuk menghindari atau mengurangi tingkat *congestion* dalam jaringan dengan cara mengatur lalu lintas jaringan. Sedangkan, *congestion management tools* digunakan untuk mengatasi *congestion* dalam jaringan berdasarkan nilai prioritas yang telah diberikan pada setiap paket.

Referensi:

- "QoS Classification and Marking," ipcisco.com. [Online]. Available: https://ipcisco.com/lesson/classification-and-marking/. [Accessed Apr. 16, 2024].
- "Quality of Service (QoS) Network Congestion Management," study-ccna.com. [Online]. Available: https://study-ccna.com/qos-network-congestion-management/. [Accessed Apr. 16, 2024].

10. Mekanisme *marking* pada *layer* 2 dan *layer* 3:

    a. *Layer* 2 (*data link layer*).

       Menggunakan mekanisme IEEE 802.1Q VLAN *tagging*. Terdapat dua *fields* yang akan

ditambahkan ke dalam *ethernet frame*, yaitu TPID (2 *bytes*) dan TCI (2 *bytes*). Dalam TCI *fields*, terdapat informasi yang menyimpan nilai prioritas dari paket tersebut (PRI *field*), terdiri dari 3 *bits* yang mengidentifikasi CoS (*Class of Service*) *markings*. Hal ini berarti bahwa nilai prioritas yang dapat diberikan pada paket tersebut berkisar antara 0 hingga 7. Untuk penjelasan dari setiap nilai prioritasnya adalah sebagai berikut:

Nilai CoS	Deksripsi
0	*Best-effort data*
1	*Medium-priority data*
2	*High-priority data*
3	*Call signalling*
4	*Video conferencing*
5	*Voice traffic*
6	Reserved
7	Reserved

b. Layer 3 (*network layer*).

Dalam *network layer*, untuk *header* paket IPv4 dan IPv6 telah menyediakan *field* untuk *marking*, yaitu *Type of Service* (ToS) dalam IPv4 dan *Traffic Class* dalam IPv6. Dalam ToS dan *traffic class* yang terdiri dari 8 *bit*, 3 *bit* diawal akan menentukan prioritas paket tersebut. 3 *bit* awal ini disebut sebagai IPP (IP *Precedence*). Namun mekanime ini sudah jarang digunakan karena belum cukup untuk menyediakan QoS yang diinginkan. Untuk mengatasi permasalahan tersebut, muncul mekanisme DSCP (*Differentiated Services Code Point*), yang menjadikan 6 *bit* awal dari ToS atau *traffic class* sebagai nilai prioritas. Nilai DSCP terbagi menjadi 3 kategori yaitu, *best-effort* (BE), *expedited forwarding* (EF), dan *assured forwarding* (AF).

Referensi:

- "QoS Marking Mechanism Explained," study-ccnp.com. [Online]. Available: https://study-ccnp.com/qos-marking-mechanism-explained/. [Accessed Apr. 16, 2024].