

# COMPUTER NETWORK MANAGEMENT

Week - 13

## Network Performance Analysis

Universitas Kristen Wira Wacana Sumba  
Lecturer - Fajar Hariadi

## Contents

- 1 **QoS**
- 2 **Parameter QoS**
- 3 **Analisis Kualitas Jaringan**

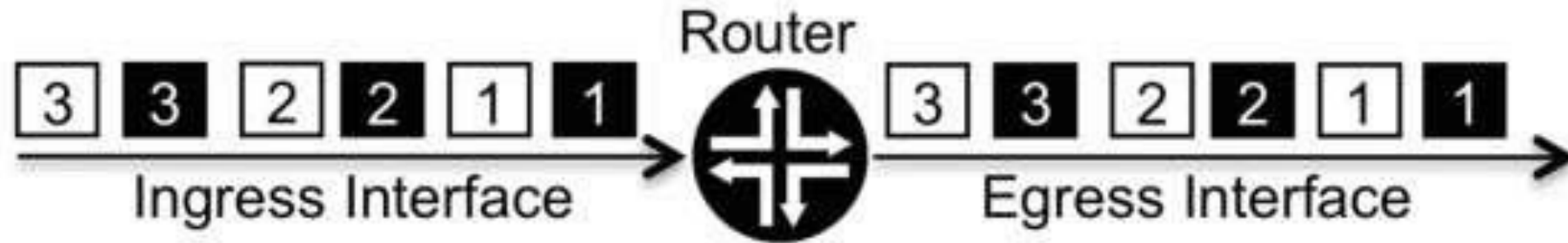
01

# Quality of Service (QoS)

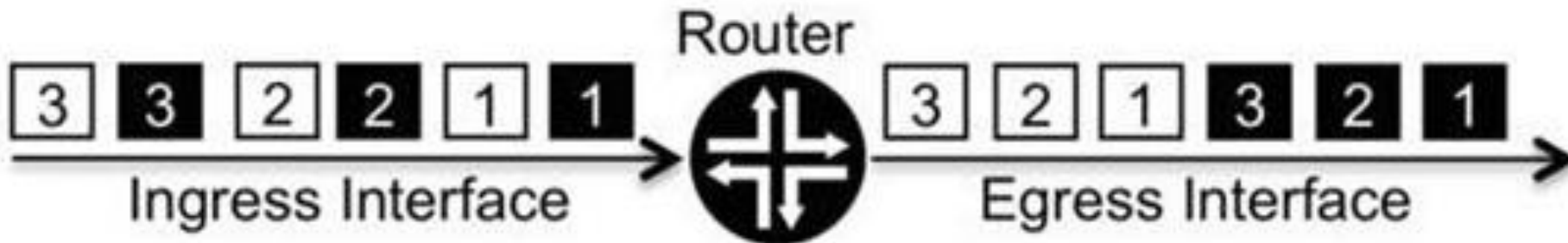
# Quality of Service (QoS)

- Quality of Service (QoS) merupakan mekanisme atau teknologi yang bekerja dengan cara mengontrol lalu lintas jaringan dan memastikan aplikasi yang penting dapat berjalan dengan lancar.
- Pemilihan aplikasi yang dianggap penting bagi organisasi tentu saja berbeda-beda, ada yang lebih mengutamakan IPTV, Voice over IP (VoIP), Video Conference, atau Game Online
- Setiap transmisi data dari aplikasi-aplikasi tersebut akan lebih didahulukan untuk diproses atau dikirimkan oleh perangkat jaringan dibandingkan data dari aplikasi lainnya

# Quality of Service (QoS)



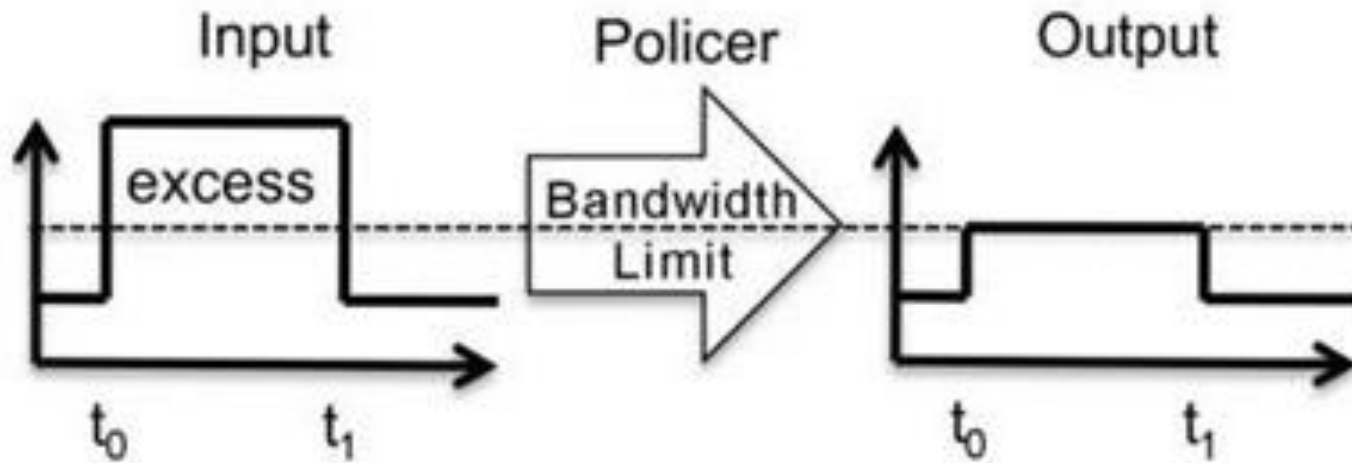
Aliran Data Tanpa QoS



Aliran Data dengan QoS

QoS bekerja dengan memprioritaskan data tertentu / penjadwalan, sehingga ada data yang lebih cepat dikirimkan dan ada data yang lebih lambat dikirimkan oleh perangkat jaringan

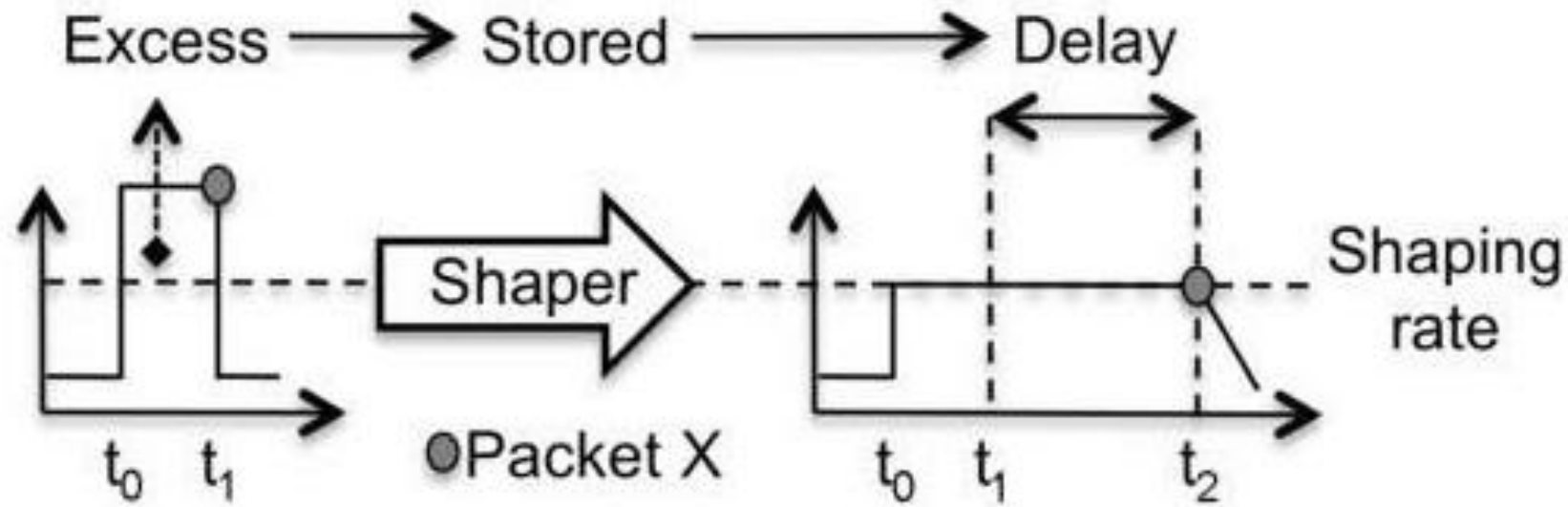
# Policer Tool



Policer tool bertugas untuk memastikan aliran data sesuai dengan batasan bandwidth yang telah ditentukan.

Hasil luaran aliran data setelah melewati policer tools adalah aliran data yang sama dengan aliran data masuk namun telah dibatasi sesuai dengan limit bandwidth yang diatur kelebihan aliran data yang melewati limit akan dibuang

# Shaper Function

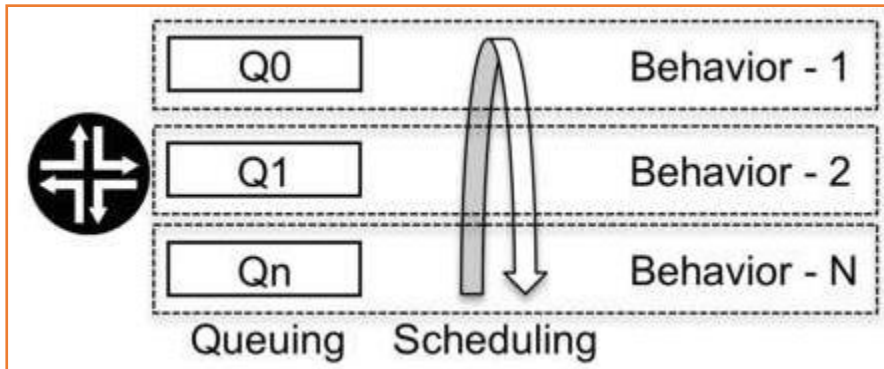


Shaper function juga bertugas untuk memastikan aliran data sesuai dengan batasan bandwidth yang telah ditentukan.

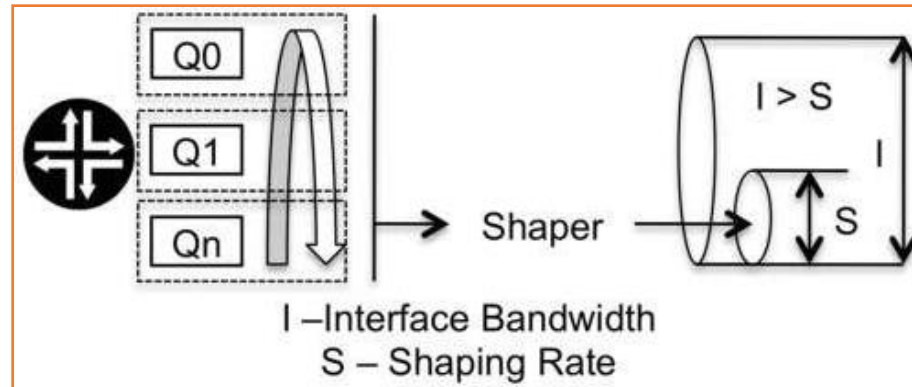
Perbedaannya adalah policer tools membuang ukuran data yang lebih dari bandwidth, sedangkan shaper function memasukkan potongan data berlebih tersebut ke dalam antrian untuk kemudian dijadwalkan untuk dikirim berikutnya, fungsi ini juga menambah delay pengiriman

# Queuing and Scheduling

Policer tool dan shaper function pada umumnya akan bekerja sama dalam fungsi antrian dan penjadwalan transmisi pada perangkat jaringan. Queue dan schedule akan diterapkan pada interface untuk dipisahkan menjadi beberapa antrian lalu kemudian scheduler akan menentukan perlakuan terhadap setiap antrian, perlakuan yang diberikan dapat saja berbeda antara antrian satu dengan yang lainnya.

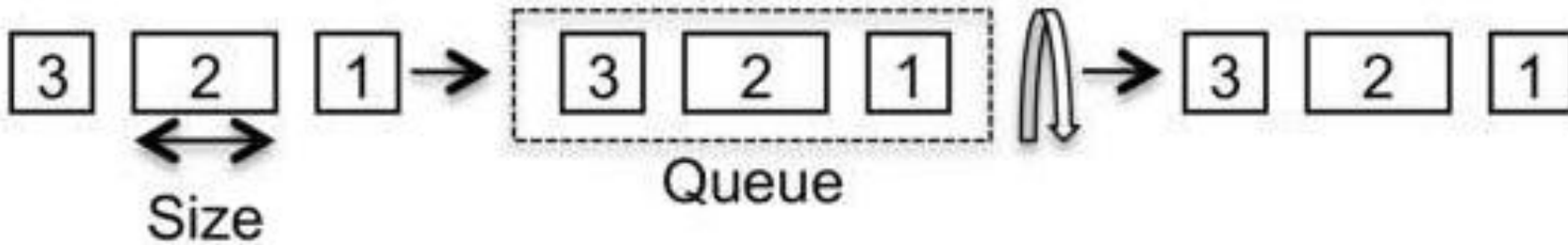


Antrian dan penjadwalan dapat menerapkan perlakuan yang berbeda



Parameter bandwidth pada antrian dan penjadwalan

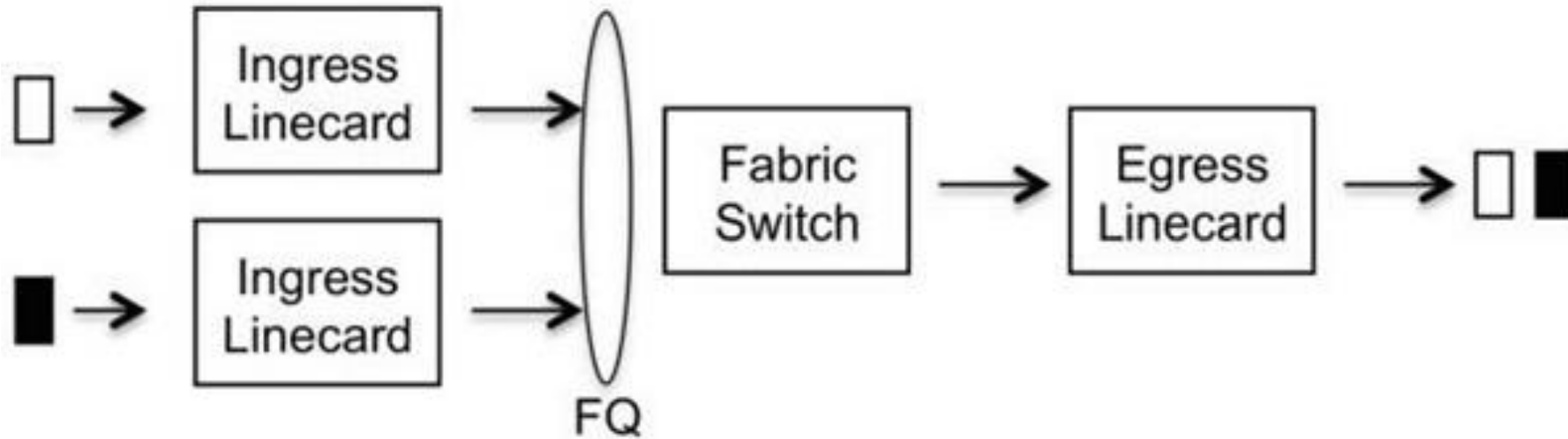
# First In First Out - FIFO



- FIFO merupakan tipe penjadwalan paling dasar. Semua paket diperlakukan sama rata sehingga siapa yang masuk duluan akan diproses dan di forward terlebih dahulu.
- Sehingga boleh dibilang tidak ada prioritas dalam antrian.
- Pada mekanisme ini berarti penanganan kelebihan paket terhadap bandwidth dihapus atau diantrikan menggunakan shaper function tergantung dari kecepatan interface atau shaper rate yang diatur pada interface.

Urutan paket yang masuk akan sama dengan urutan paket yang keluar karena bentuk antrian Cuma satu dan tidak ada mekanisme perubahan sistem antrian

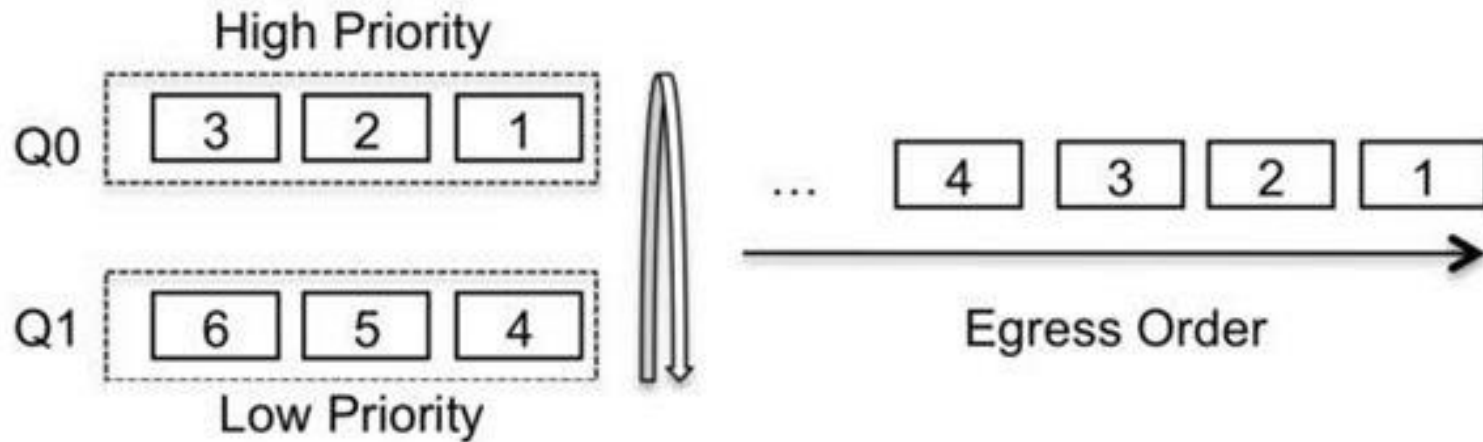
# Fair Queuing



Fair queuing (FQ) menggunakan algoritma fairness yang merupakan penjadwalan dengan membagi aliran data utama menjadi beberapa antrian.

- Antrian-antrian yang dibuat kemudian di forward secara bergantian dengan bobot yang sama.
- FQ memisahkan antara lalu lintas jaringan secara keseluruhan dengan aliran data per aplikasi.
- Hal ini digunakan untuk menghindari aplikasi yang memerlukan sedikit bandwidth tertutupi oleh aliran data aplikasi yang memerlukan aliran data yang besar

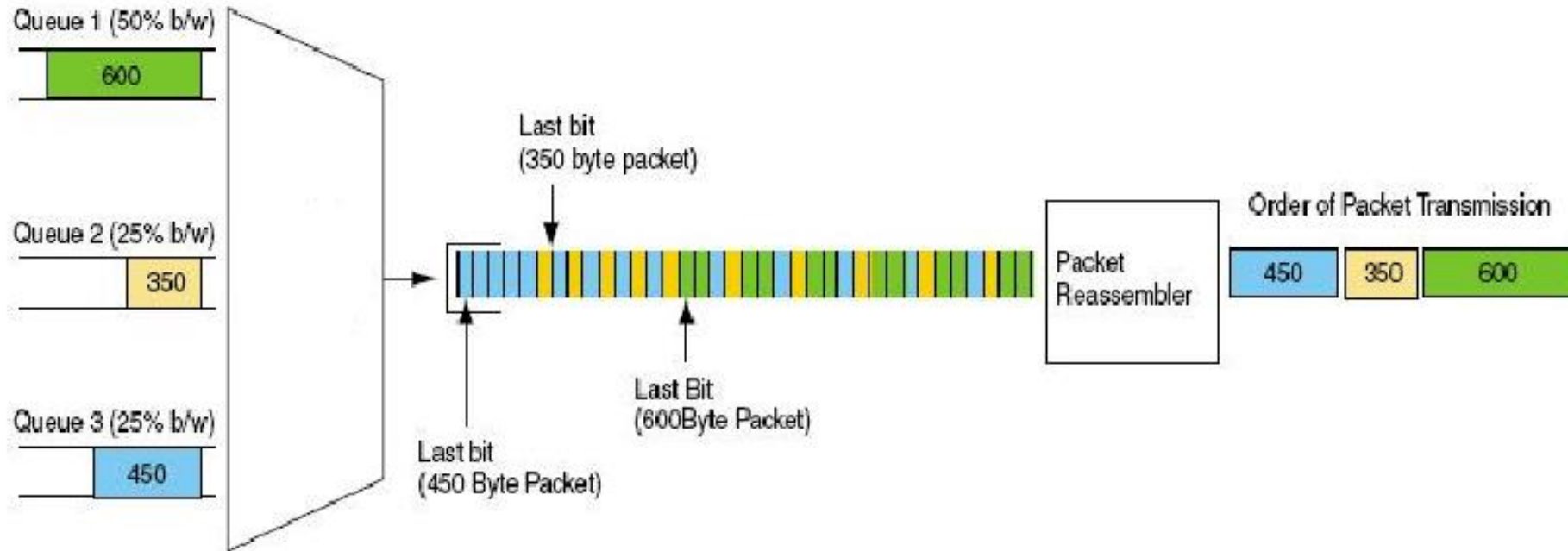
# Priority Queuing



Priority queuing biasa di pakai pada jaringan yang memiliki aplikasi yang sensitif terhadap delay dan packet loss, sehingga pemrosesan dan penjadwalan akan diprioritaskan untuk aplikasi tersebut

- PQ memiliki kemampuan untuk membagi kelas berdasarkan skema tertentu
- Setelah dibagi kelas paket akan ditempatkan ke dalam antrian yang berbeda dan diberikan prioritas yang berbeda.
- Antrian yang memiliki prioritas lebih tinggi akan didahulukan untuk dikirim, sampai antrian kosong baru kemudian dilanjutkan untuk antrian yang lebih rendah prioritasnya

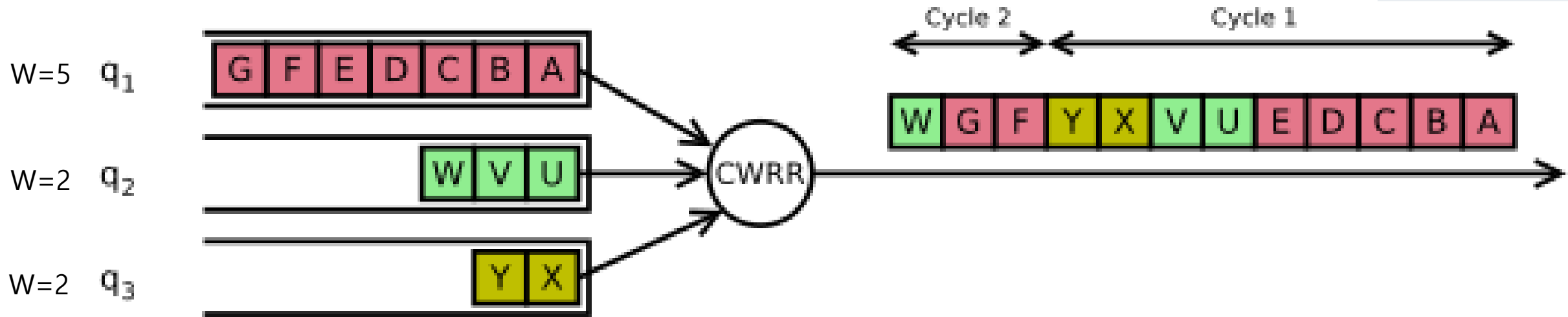
# Weighted Fair Queuing



Weighted fair queuing sering juga disebut bit by bit round robin karena mekanisme antrian dan penjadwalan berdasarkan urutan dari bit-bit paket antriannya

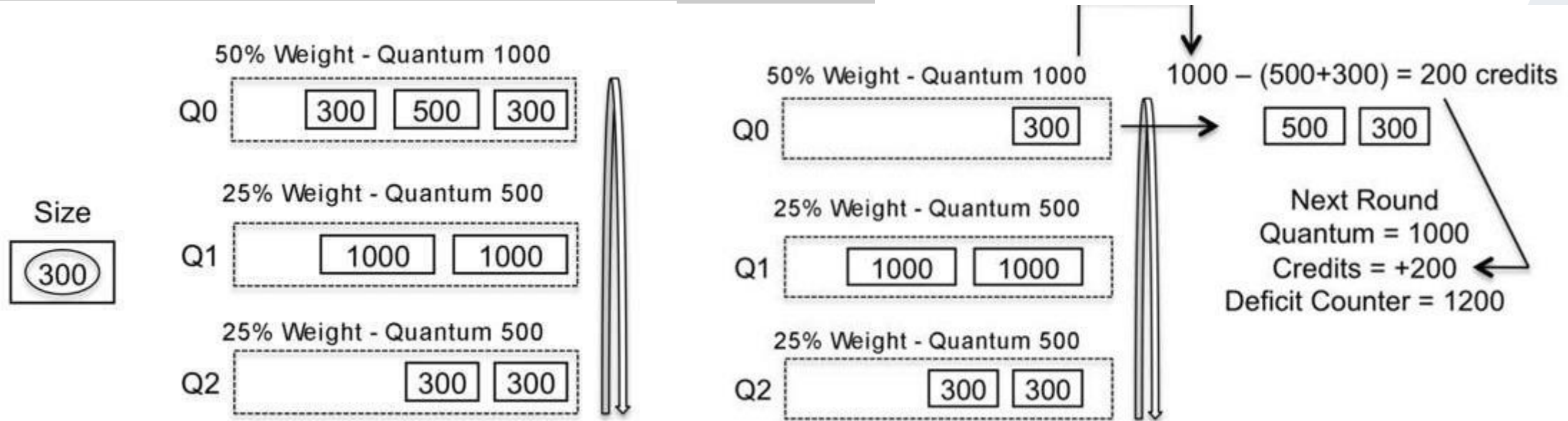
- Pada WFQ paket-paket akan diurutkan sesuai dengan antriannya
- Selanjutnya antrian dengan prioritas tertinggi akan diberikan kesempatan lebih lama untuk mengirimkan pesan per bit-nya dibandingkan dengan antrian yang prioritasnya rendah
- Sampai pada bit terakhir setiap paket dalam antrian bit-bit paket akan disusun kembali untuk dikirimkan paket seutuhnya.

# Weighted Round Robin



- Pada WRR paket diklasifikasikan ke beberapa antrian dan diberikan bobot
- Pada putaran pertama  $q_1$  karena bobotnya 5 maka paket yang dikirim adalah 5, sisa paket yang lainnya akan menunggu di putaran berikutnya
- Sedangkan  $q_2$  karena bobotnya 2 maka 2 paket akan dikirimkan, sisanya menunggu putaran berikutnya
- Pada  $q_3$  karena bobotnya 2 maka seluruh antrian dikirimkan
- Pada putaran kedua seluruh paket  $q_1$  dikirimkan diikuti dengan sisa paket pada  $q_2$

# Deficit Weighted Round Robin



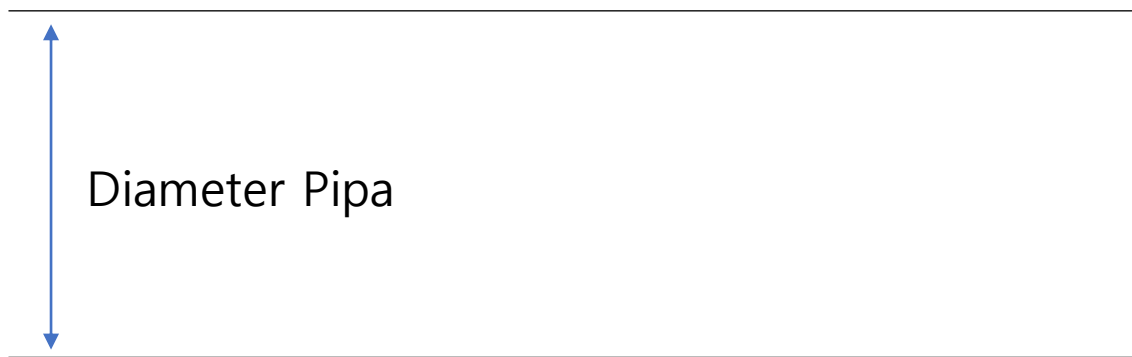
- Deficit WRR merupakan modifikasi WRR dengan menambahkan penjadwalan dengan mempertimbangkan variasi dari besarnya paket
- Pada deficit round robin setiap antrian diberikan kredit, bila besar paket lebih kecil dari kredit maka paket akan dikirimkan dan sisa kreditnya akan disimpan untuk putaran berikutnya
- Paket yang lebih besar dari kredit akan disimpan dan nilai kreditnya akan ditambahkan ke putaran berikutnya, sehingga ketika nilai kredit sudah mencapai besarnya paket, baru paket akan dikirimkan

02

# Parameter QoS

# Bandwidth

- Bandwidth adalah luas atau lebar cakupan frekuensi yang digunakan oleh sinyal dalam medium transmisi. Bandwidth sering digunakan sebagai suatu sinonim untuk kecepatan transfer data (transfer rate) yaitu jumlah data yang dapat dibawa dari sebuah titik ke titik lain dalam jangka waktu tertentu (pada umumnya dalam detik).
- Besarnya bandwidth bergantung dari kualitas perangkat yang kita gunakan atau kalau dalam hal langganan internet bergantung dari besarnya paket internet yang dibayarkan



Bandwidth dapat digambarkan seperti pipa air

# Throughput

- Throughput adalah kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Biasanya throughput selalu dikaitkan dengan bandwidth dalam kondisi yang sebenarnya. Bandwidth lebih bersifat fix sementara throughput sifatnya adalah dinamis tergantung trafik yang sedang terjadi.
- Beberapa faktor yang mempengaruhi bandwidth dan throughput yaitu antara lain piranti jaringan, tipe data yang ditransfer, banyaknya pengguna jaringan, topologi jaringan, spesifikasi computer client/user, spesifikasi server komputer, induksi listrik, cuaca dan lain sebagainya.

---

Banyaknya Air = throughput

Throughput dapat digambarkan seperti banyaknya air yang mengalir melewati pipa dalam satuan waktu

# Throughput

- Throughput adalah kecepatan (rate) transfer data efektif yang diukur dalam bps. Throughput merupakan jumlah total kedatangan paket yang sukses yang diamati pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. Throughput dirumuskan :

$$\text{Throughput} : \frac{\text{Packed received (kb)}}{\text{Time transmitted (s)}}$$

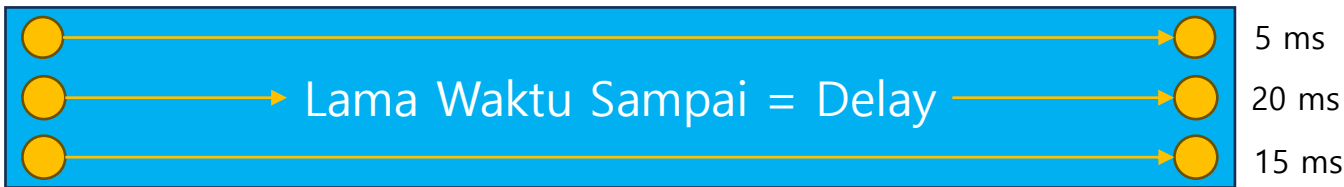
Standar Throughput menurut TIPHON

<b>Kategori Throughput</b>	<b>Throughput</b>	<b>Indeks</b>
<i>Bad</i>	0 – 338 kbps	0
<i>Poor</i>	338 – 700 kbps	1
<i>Fair</i>	700 – 1200 kbps	2
<i>Good</i>	1200 kbps – 2,1 Mbps	3
<i>Excelent</i>	>2,1 Mbps	4

# Delay

- Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, kongesti atau waktu proses yang lama.

$$\text{Delay rata-rata} = \frac{\text{total delay}}{\text{Jumlah paket yang diterima}}$$



Delay seperti rata-rata waktu sampai setiap elemen air ke tujuan

Standar Delay menurut TIPHON

<b>Delay (ms)</b>	<b>Indeks</b>	<b>Kategori Delay</b>
<150	4	Sangat Bagus
150 s/d 300	3	Bagus
300 s/d 450	2	Sedang
>450	1	Jelek

# Jitter

- Jitter adalah variasi atau perubahan latency dari delay atau variasi waktu kedatangan paket.
- Banyak hal yang dapat menyebabkan jitter, antara lain:
  - Panjangnya antrian dalam waktu pengolahan data,
  - Peningkatan trafik secara tiba-tiba sehingga menyebabkan penyempitan bandwidth dan menimbulkan antrian dan,
  - Kecepatan terima dan kirim paket dari setiap node juga dapat menyebabkan jitter.

$$\text{Jitter rata - rata} = \frac{\text{Total variasi delay}}{\text{total paket yang diterima}}$$

Standar Jitter menurut TIPHON

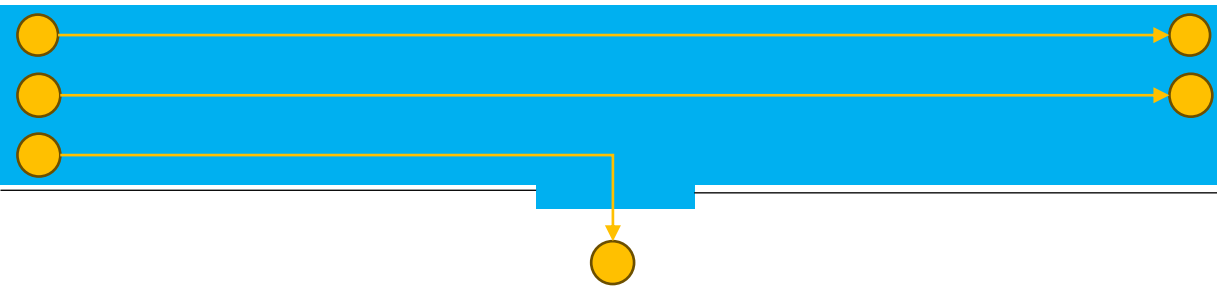
Kategori Jitter	Jitter	Indeks
<i>Poor</i>	125 – 225 ms	1
<i>Medium</i>	75 – 125 ms	2
<i>Good</i>	0 – 75 ms	3
<i>Perfect</i>	0 ms	4



Jitter merupakan variasi selisih waktu sampai antar elemen

# Packet Loss

- Packet loss adalah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang.
- Packet Loss dapat disebabkan oleh beberapa hal:
  - Terjadinya overload trafik didalam jaringan.
  - Tabrakan (congestion) dalam jaringan.
  - Error yang terjadi pada media fisik.
  - Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena Overflow yang terjadi pada buffer.



Packet loss merupakan elemen/paket yang tidak sampai ke tujuan

$$\text{Packet loss} = \frac{(\text{Packet transmitted} - \text{Packet received})}{\text{Packet transmitted}} \times 100\%$$

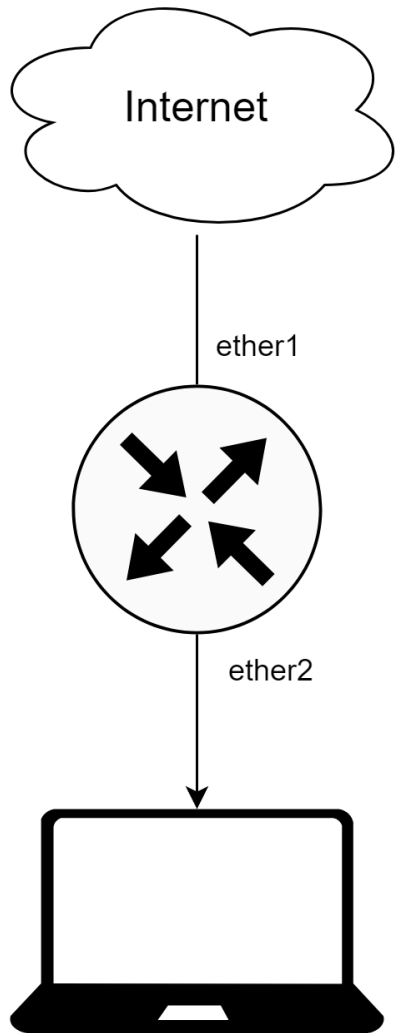
Standar Packet Loss menurut TIPHON

<b>Kategori Packet Loss</b>	<b>Packet Loss</b>	<b>Indeks</b>
<i>Poor</i>	>25%	1
<i>Medium</i>	12 – 24%	2
<i>Good</i>	3 – 14%	3
<i>Perfect</i>	0 – 2%	4

03

# Analisis Kualitas Jaringan

# Topologi



Bridge	Port	DHCP	IP Address
Internet	ether1	DHCP Client	-
LAN	ether2, ether3	DHCP Server	192.168.10.1/24

- Ether1 akan masuk dalam bridge Internet yang merupakan sumber internet
- Ether2 dan Ether 3 masuk dalam bridge LAN, dan bridge ini akan menjadi DHCP Server bagi setiap client yang terhubung dengan Ether2 atau Ether3

# Membuat Bridge - Router

## Langkah Konfigurasi Bridge

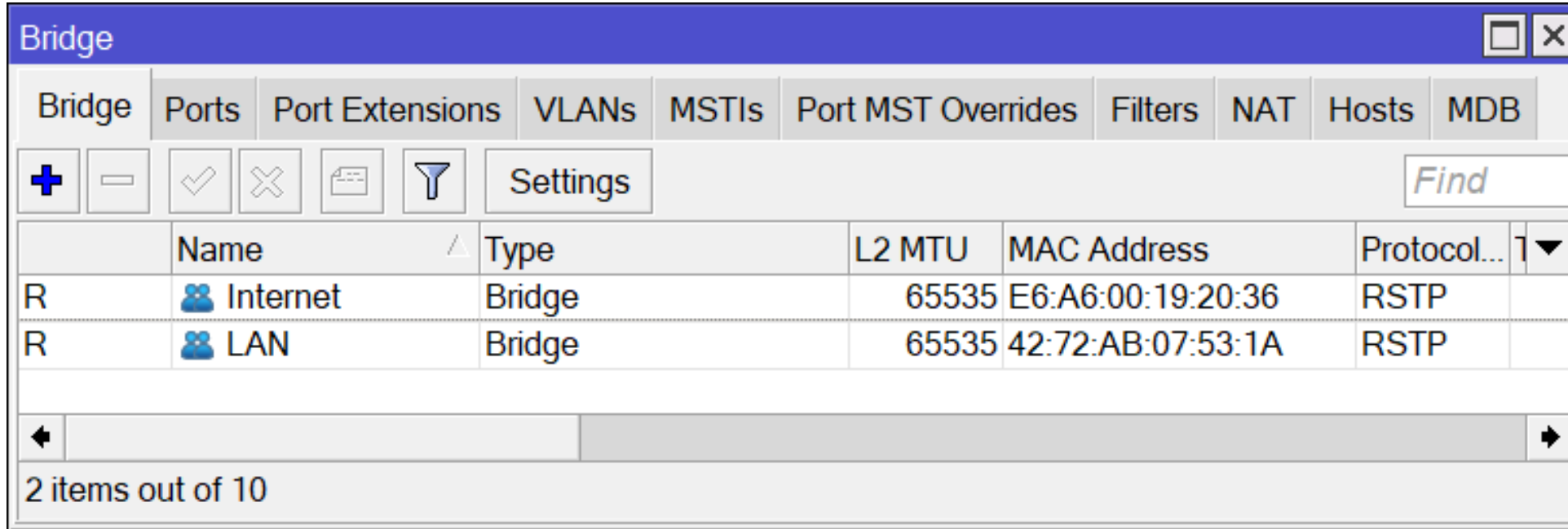
The screenshot shows the MikroTik WinBox interface. On the left, the 'Bridge' menu is highlighted with a red box and a '1'. In the center, the 'Bridge' window shows a table with columns 'Name', 'Type', 'L2 MTU', 'MAC Address', and 'Protocol...'. A red box with a '2' highlights the '+' icon in the top-left corner of the table. On the right, the 'New Interface' dialog is open. The 'Name' field is set to 'Internet' and is highlighted with a red box and a '3'. The 'Type' is set to 'Bridge'. The 'OK' button is highlighted with a red box and a '4'. The status bar at the bottom shows 'enabled', 'running', 'slave', and 'passthrough'.

Buat bridge untuk kedua sesuai dengan tabel

Bridge	Port	DHCP	IP Address
Internet	ether1	DHCP Client	-
LAN	ether2, ether3	DHCP Server	192.168.10.1/24

# Membuat Bridge - Router

Hasil bridge akan terlihat seperti gambar berikut



The screenshot shows a window titled "Bridge" with a menu bar containing "Bridge", "Ports", "Port Extensions", "VLANs", "MSTIs", "Port MST Overrides", "Filters", "NAT", "Hosts", and "MDB". Below the menu bar is a toolbar with icons for adding (+), removing (-), checking (✓), unchecking (✗), a document icon, a funnel icon, and a "Settings" button. A search box labeled "Find" is also present. The main area contains a table with the following data:

	Name	Type	L2 MTU	MAC Address	Protocol...	
R	Internet	Bridge	65535	E6:A6:00:19:20:36	RSTP	
R	LAN	Bridge	65535	42:72:AB:07:53:1A	RSTP	

At the bottom of the window, it says "2 items out of 10".

Selanjutnya memasukkan port sebagai anggota bridge sesuai dengan tabel berikut:

Bridge	Port	DHCP	IP Address
Internet	ether1	DHCP Client	-
LAN	ether2, ether3	DHCP Server	192.168.10.1/24

# Port Bridge - Router

Untuk memasukkan port menjadi anggota bridge, masuk ke dalam tab port

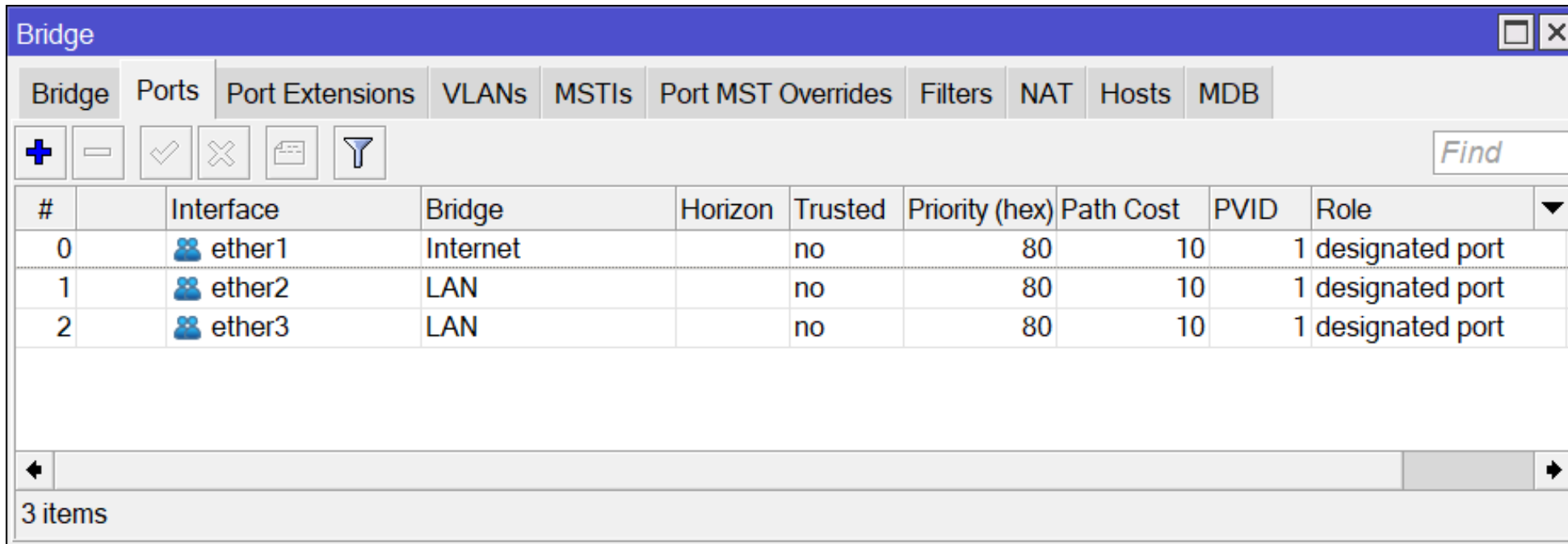
The image shows two screenshots from Mikrotik WinBox. The left screenshot shows the 'Bridge' configuration window with the 'Ports' tab selected. A red box labeled '1' highlights the 'Ports' tab, and another red box labeled '2' highlights the '+' icon in the toolbar. The right screenshot shows the 'New Bridge Port' dialog box. A red box labeled '3' highlights the 'Interface' and 'Bridge' dropdown menus, and another red box labeled '4' highlights the 'OK' button.

Tambahkan semua port pada bridge yang sesuai

Bridge	Port	DHCP	IP Address
Internet	ether1	DHCP Client	-
LAN	ether2, ether3	DHCP Server	192.168.10.1/24

# Port Bridge - Router

Hasil penambahan port pada bridge yang telah dibuat:



The screenshot shows a network configuration window titled "Bridge" with a tabbed interface. The "Ports" tab is selected. Below the tabs are several icons for adding, deleting, and filtering ports, along with a "Find" search box. The main area contains a table with the following columns: #, Interface, Bridge, Horizon, Trusted, Priority (hex), Path Cost, PVID, and Role. There are three rows of data in the table.

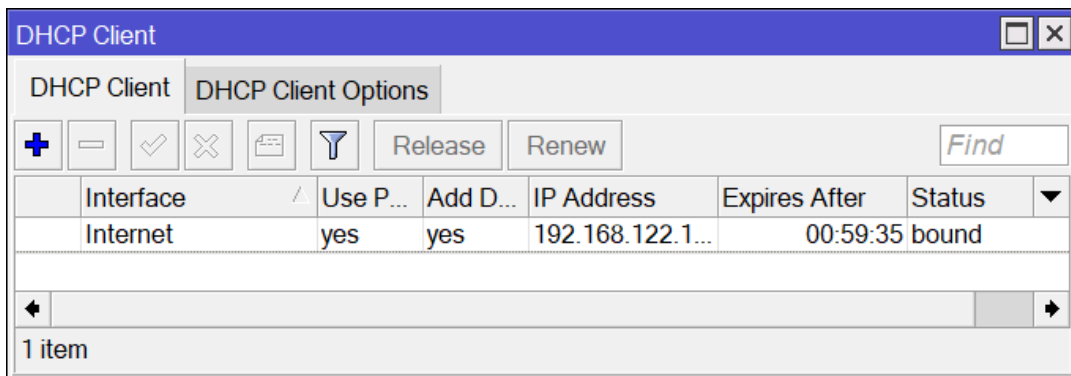
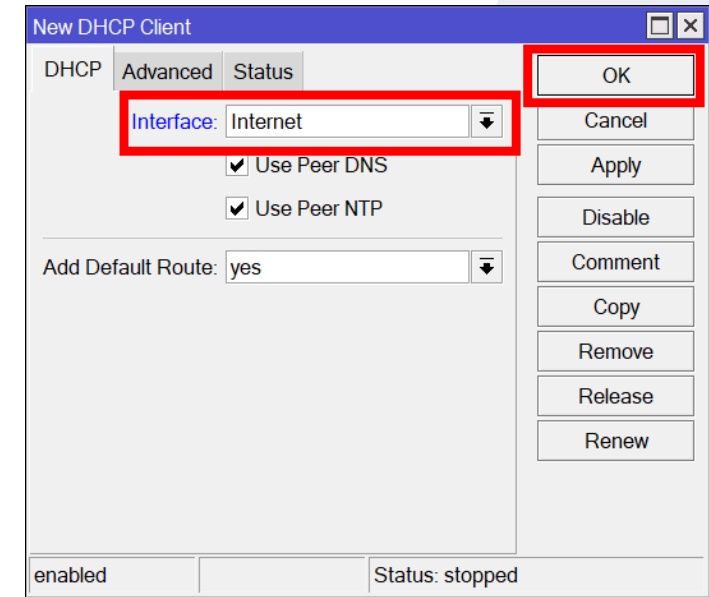
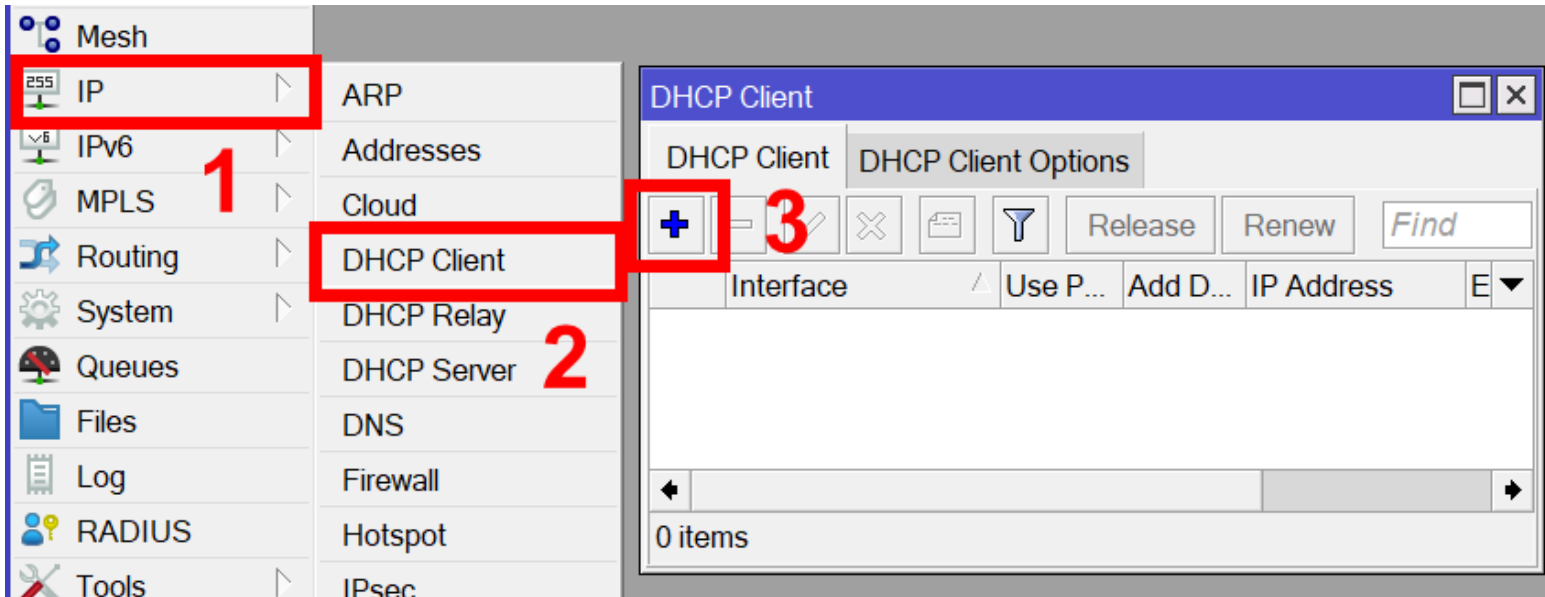
#	Interface	Bridge	Horizon	Trusted	Priority (hex)	Path Cost	PVID	Role
0	ether1	Internet		no	80	10	1	designated port
1	ether2	LAN		no	80	10	1	designated port
2	ether3	LAN		no	80	10	1	designated port

3 items

Bridge	Port	DHCP	IP Address
Internet	ether1	DHCP Client	-
LAN	ether2, ether3	DHCP Server	192.168.10.1/24

# DHCP Client - Router

IP address untuk koneksi internet perlu didapatkan pada bridge internet yang telah dibuat, dengan cara menjadikan bridge sebagai DHCP Client



Hasil IP Address yang didapat bisa saja berbeda, tergantung dari pengaturan sumber internet atau langganan ISP

# IP Address Gateway - Router

Sebelum dapat membuat DHCP Server yang harus dilakukan adalah menentukan IP Address interface pada router yang akan menjadi DHCP Server

Pada list yang ada sudah terdapat ip address dari DHCP Client sebelumnya

The screenshot shows the Mikrotik WinBox interface. On the left, the configuration tree is visible with the following items highlighted by red boxes and numbers:

- 1**: IP
- 2**: Addresses
- 3**: A plus sign icon (+) used for adding new entries.

The main window displays the 'Address List' table, which contains one entry:

	Address	Network	Interface
D	192.168.122.1...	192.168.122.0	Internet

The status '1 item' is shown at the bottom of the table.

# IP Address Gateway - Router

IP Address pada bridge LAN ini merupakan gateway yang akan digunakan oleh seluruh komputer yang terhubung dengan interface bridge LAN (ether2, ether3)

New Address

Address: 192.168.10.1/24

Network: [dropdown]

Interface: LAN [dropdown]

OK

Cancel

Apply

Disable

Comment

Copy

Remove

enabled

1

2

Hasil pemberian IP Address gateway terlihat seperti berikut

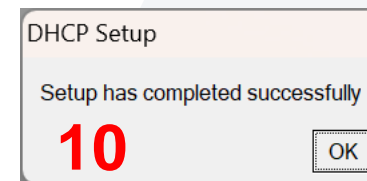
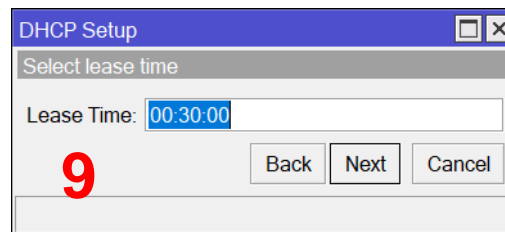
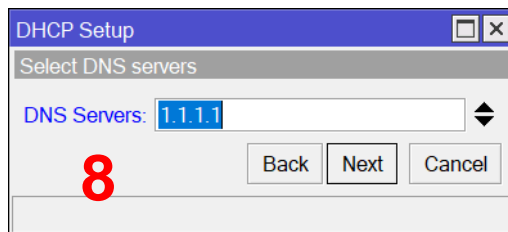
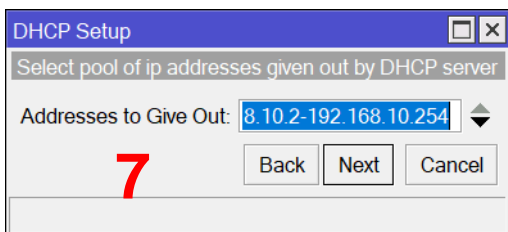
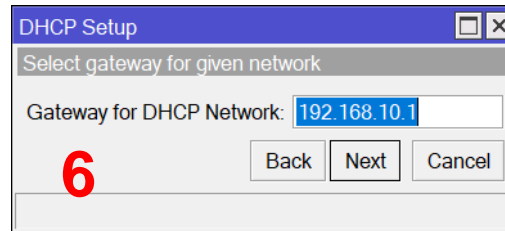
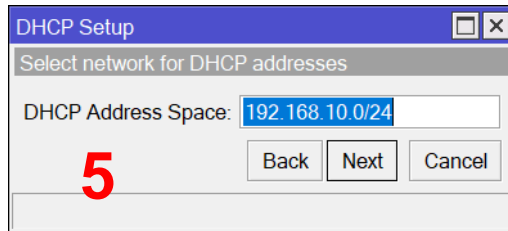
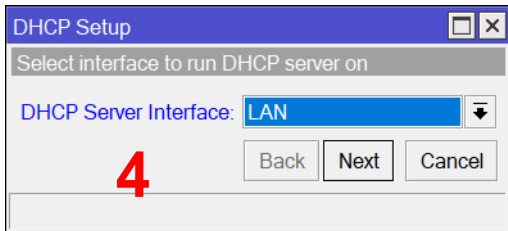
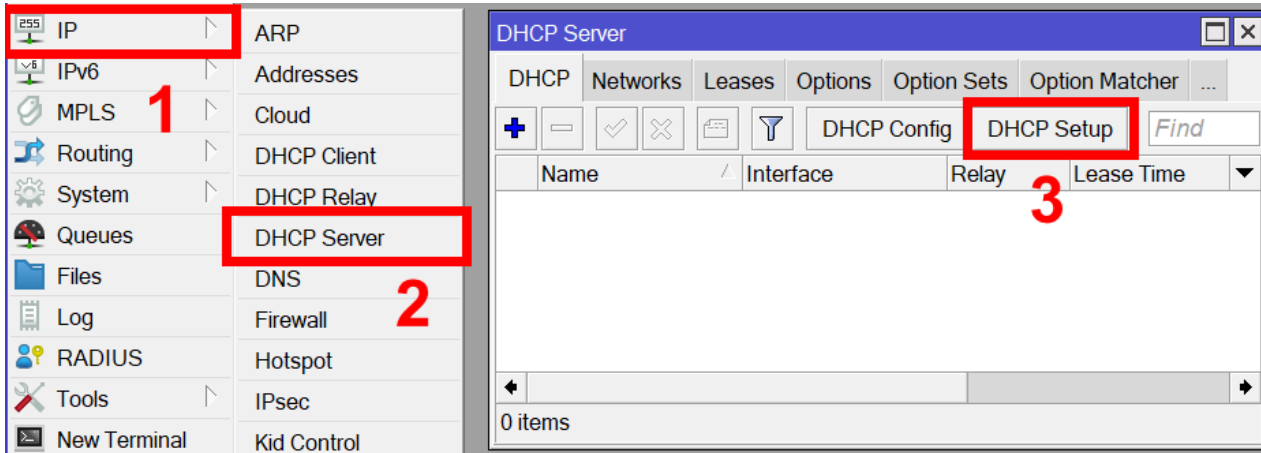
Address List

	Address	Network	Interface
	192.168.10.1/24	192.168.10.0	LAN
D	192.168.122.1...	192.168.122.0	Internet

2 items

# DHCP Server - Router

Setelah sudah terdapat gateway selanjutnya dapat dibuat dhcp server untuk bridge LAN, sehingga setiap client yang terhubung dapat IP Address secara otomatis



# DHCP Server - Router

Hasil DHCP Server akan terlihat seperti gambar berikut:

Name	Interface	Relay	Lease Time	Address Pool	Add AR...
dhcp1	LAN		00:30:00	dhcp_pool0	no

Setiap komputer yang terhubung dengan interface bridge LAN akan mendapatkan ip address dari dhcp server ini, sehingga memiliki rentang ip address pada network 192.168.10.0/24

# NAT - Router

Pembuatan NAT dilakukan pada menu firewall pada winbox

The image shows the Mikrotik WinBox interface. On the left, the 'System' menu is expanded, and 'Firewall' is highlighted with a red box and the number '2'. The 'IP' menu item is also highlighted with a red box and the number '1'. On the right, the 'Firewall' configuration window is open, with the 'NAT' tab selected and highlighted with a red box and the number '3'. The '+' icon for adding a new rule is highlighted with a red box and the number '4'. The table below shows the columns for rule configuration.

#	Action	Chain	Src. Address	Dst. Address	Src. Ad...	Dst. Ad...	Proto...	Src. Port
0 items								

# NAT - Router

Chain: srcnat

Out. Interface: Internet

enabled

Chain srcnat dan out interface Internet artinya pengelompokan setiap traffic yang bersumber dari jaringan local menuju ke internet

Action: masquerade

Log Prefix:

enabled

Action masquerade dari setiap traffic yang sudah dikelompokkan tersebut akan ditranslasikan menjadi agar terlihat bersumber dari ip address public di interface bridge internet untuk kemudian dilanjutkan ke jaringan di internet

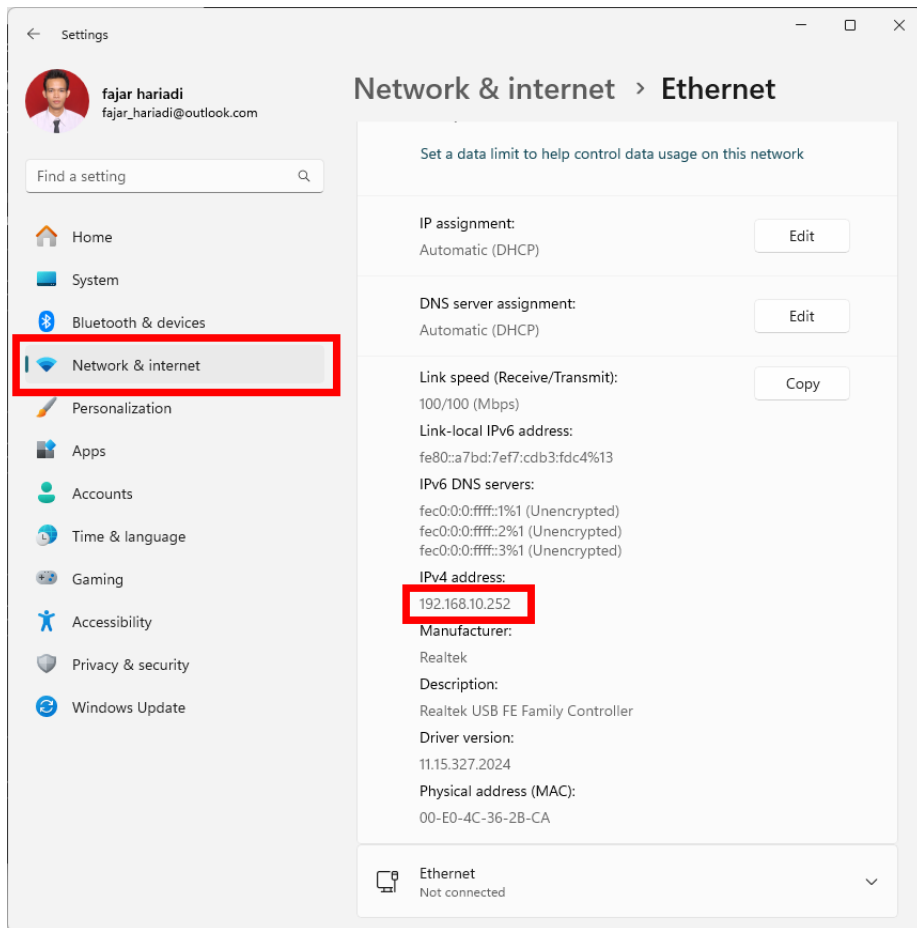
Hasil nat

#	Action	Chain	Src. Address	Dst. Address	Src. Ad...	Dst. Ad...	Proto...	Src. Port	Dst. Port	In. Inter...	Out. Int...	In. Inter...	Out. Int...	Bytes
0	masquerade	srcnat									Internet			0 B

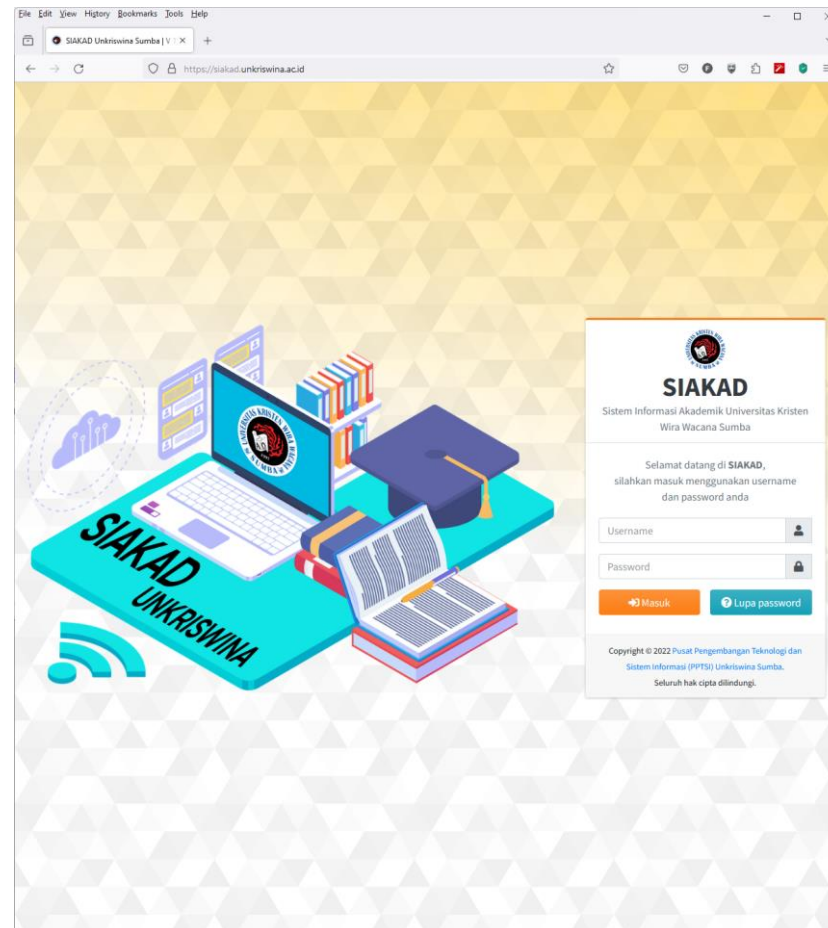
1 item

# Pengujian Koneksi

## IP Address



## Koneksi Internet



IP Address yang didapat berasal dari DHCP Server Mikrotik-1 walaupun Laptop/PC terhubung dengan mikrotik -2

Koneksi internet juga bisa didapat

# Capture Packet

Buka aplikasi wireshark, lalu pilih interface yang terkoneksi ke router, lalu start capture, dan buka beberapa tab pada browser

The screenshot shows the Wireshark Network Analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for file operations and analysis. The main window displays a 'Welcome to Wireshark' message and a list of 'Open' files. Below this, the 'Capture' panel is visible, showing a list of network interfaces. The 'Ethernet 2' interface is highlighted with a red box, and a red number '1' is placed above it. The status bar at the bottom indicates 'Ready to load or capture', 'No Packets', and 'Profile: Default'.

2

Welcome to Wireshark

Open

- C:\Users\fajar\Documents\sebelum manajemen bandwidth.pcapng (16 MB)
- C:\Users\fajar\Documents\setelah manajemen bandwidth.pcapng (7449 KB)
- C:\Users\fajar\Documents\Sebelum (19 MB)
- C:\Users\fajar\Downloads\Data-Sniffer1.unknown (10000 KB)

1

Capture

...using this filter:  12 interfaces shown, 12 hidden

- Ethernet 2**
- Local Area Connection
- Adapter for loopback traffic capture
- Local Area Connection\* 12
- Local Area Connection\* 11
- Local Area Connection\* 10
- Wi-Fi
- Local Area Connection\* 4
- Local Area Connection\* 3
- Ethernet 3
- Ethernet
- Event Tracing for Windows (ETW) reader

Learn

[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#) · [SharkFest](#) · [Wireshark Discord](#) · [Donate](#)

You are running Wireshark 4.2.5 (v4.2.5-0-g4aa814ac25a1). You receive automatic updates.

Ready to load or capture || No Packets || Profile: Default

# Capture Packet

Wireshark akan mengcapture aliran data, tunggu beberapa saat baru stop untuk dianalisis lebih lanjut kualitas jaringannya (Parameter QoS)

The screenshot shows the Wireshark interface with a packet capture from Ethernet 2. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, and Protocol. The packets are color-coded: black for packets that did not reach the destination and light blue for those that did. The bottom pane shows the details of a selected packet (No. 6000), including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Transport Layer Security.

No.	Time	Source	Destination	Protocol
5968	0.000026	192.168.10.254	149.50.211.10	TCP
5969	0.000811	192.168.10.254	149.50.211.10	TLSv1.2
5970	0.000483	192.168.10.254	149.50.211.10	TLSv1.2
5971	0.000957	192.168.10.254	149.50.211.10	TLSv1.2
5972	0.004842	192.168.10.254	149.50.211.10	TLSv1.2
5973	0.001981	149.50.211.10	192.168.10.254	TLSv1.2
5974	0.000000	149.50.211.10	192.168.10.254	TLSv1.2
5975	0.000279	192.168.10.254	149.50.211.10	TLSv1.2
5976	0.001233	192.168.10.254	149.50.211.10	TLSv1.2
5977	0.000240	192.168.10.254	149.50.211.10	TLSv1.2
5978	0.005457	149.50.211.10	192.168.10.254	TCP
5979	0.000000	149.50.211.10	192.168.10.254	TCP
5980	0.000041	192.168.10.254	149.50.211.10	TCP
5981	0.000749	149.50.211.10	192.168.10.254	TCP
5982	0.000000	149.50.211.10	192.168.10.254	TLSv1.2
5983	0.000028	192.168.10.254	149.50.211.10	TCP
5984	0.006064	192.168.10.254	149.50.211.10	TLSv1.2
5985	0.000682	192.168.10.254	149.50.211.10	TLSv1.2
5986	0.000816	149.50.211.10	192.168.10.254	TCP
5987	0.000000	149.50.211.10	192.168.10.254	TCP
5988	0.000031	192.168.10.254	149.50.211.10	TCP
5989	0.003284	192.168.10.254	149.50.211.10	TLSv1.2
5990	0.005266	149.50.211.10	192.168.10.254	TCP
5991	0.000000	149.50.211.10	192.168.10.254	TLSv1.2
5992	0.000034	192.168.10.254	149.50.211.10	TCP
5993	0.006057	192.168.10.254	149.50.211.10	TLSv1.2
5994	0.000662	192.168.10.254	149.50.211.10	TLSv1.2
5995	0.004518	192.168.10.254	149.50.211.10	TLSv1.2
5996	0.006607	149.50.211.10	192.168.10.254	TCP
5997	0.000000	149.50.211.10	192.168.10.254	TCP
5998	0.000000	149.50.211.10	192.168.10.254	TCP
5999	0.000000	149.50.211.10	192.168.10.254	TCP
6000	0.000000	149.50.211.10	192.168.10.254	TCP
6001	0.000000	149.50.211.10	192.168.10.254	TCP
6002	0.000050	192.168.10.254	149.50.211.10	TCP
6003	0.000474	149.50.211.10	192.168.10.254	TCP
6004	0.000000	149.50.211.10	192.168.10.254	TCP
6005	0.000000	149.50.211.10	192.168.10.254	TCP
6006	0.000631	149.50.211.10	192.168.10.254	TCP
6007	0.008286	192.168.10.254	149.50.211.10	TCP
6008	0.000266	149.50.211.10	192.168.10.254	TCP
6009	0.011013	149.50.211.10	192.168.10.254	TCP

Frame 1: 374 bytes on wire (2992 bits), 374 bytes captured (2992 bits) on interface 0  
Ethernet II, Src: Routerboardc\_98:d0:79 (6c:3b:6b:98:d0:79), Dst: 08:00:00:00:00:00  
> Internet Protocol Version 4, Src: 149.50.211.10, Dst: 192.168.10.254  
> Transmission Control Protocol, Src Port: 443, Dst Port: 56615, Seq: 123456789  
> Transport Layer Security

Paket berwarna hitam artinya merupakan paket yang tidak sampai ke tujuan, sedangkan warna lainnya merupakan paket yang berhasil sampai ke tujuan, hanya dibedakan berdasarkan protokol yang digunakan

# Capture Packet

Filter datanya agar hanya paket yang tidak sampai saja yang terlihat dengan menggunakan filter:  
tcp.analysis.out\_of\_order || tcp.analysis.retransmission || tcp.analysis.duplicate\_ack ||  
tcp.analysis.lost\_segment

The screenshot shows the Wireshark interface with a packet capture of an Ethernet II frame. The filter bar at the top contains the filter: `tcp.analysis.out_of_order || tcp.analysis.retransmission || tcp.analysis.duplicate_ack || tcp.analysis.lost_segment`. The packet list pane shows a series of packets, with many marked as retransmissions or out of order. The packet details pane shows the structure of a TCP segment, including the source and destination ports, sequence number, and acknowledgment number.

No.	Time	Source	Destination	Protocol
65	0.038760	149.50.211.10	192.168.10.254	[TCP Dup ACK 60#1] 443 → 56616 [ACK] Seq=1 Ack=1399 Win=8923 Len=0 SLE=1632 SRE=1987
101	0.039818	149.50.211.10	192.168.10.254	[TCP Dup ACK 84#1] 443 → 56616 [ACK] Seq=1 Ack=2793 Win=8966 Len=0 SLE=3026 SRE=3221
192	0.000000	149.50.211.10	192.168.10.254	[TCP Dup ACK 191#1] 443 → 56618 [ACK] Seq=1 Ack=234 Win=17498 Len=0
194	0.000059	192.168.10.254	149.50.211.10	[TCP Spurious Retransmission] 56618 → 443 [PSH, ACK] Seq=234 Ack=1 Win=514 Len=260
195	0.036649	149.50.211.10	192.168.10.254	[TCP Dup ACK 193#1] 443 → 56618 [ACK] Seq=1 Ack=494 Win=17507 Len=0
237	0.001689	149.50.211.10	192.168.10.254	[TCP Dup ACK 229#1] 443 → 56618 [ACK] Seq=1 Ack=2542 Win=17571 Len=0 SLE=2775 SRE=3050
522	0.024778	149.50.211.10	192.168.10.254	[TCP Dup ACK 521#1] 443 → 56615 [ACK] Seq=12259 Ack=9558 Win=15564 Len=0 SLE=2701 SRE=2986
633	0.036390	149.50.211.10	192.168.10.254	[TCP Dup ACK 579#1] 443 → 56615 [ACK] Seq=18586 Ack=6388 Win=19681 Len=0 SLE=6539 SRE=6686
798	0.005564	149.50.211.10	192.168.10.254	[TCP Dup ACK 797#1] 443 → 56615 [ACK] Seq=35954 Ack=14367 Win=15933 Len=0 SLE=14567 SRE=14825
823	0.000000	149.50.211.10	192.168.10.254	[TCP Dup ACK 805#1] 443 → 56615 [ACK] Seq=44157 Ack=15664 Win=15973 Len=0 SLE=15897 SRE=16140
870	0.001738	149.50.211.10	192.168.10.254	[TCP Dup ACK 838#1] 443 → 56615 [ACK] Seq=45095 Ack=17732 Win=16038 Len=0 SLE=17965 SRE=18352
993	0.003289	149.50.211.10	192.168.10.254	[TCP Previous segment not captured] 443 → 56616 [ACK] Seq=2801 Ack=5652 Win=9056 Len=1400 [TCP segment of a reassembled PDU]
996	0.000059	192.168.10.254	149.50.211.10	[TCP Dup ACK 2#1] 56616 → 443 [ACK] Seq=5652 Ack=1 Win=514 Len=0 SLE=2801 SRE=5601
1002	0.000767	149.50.211.10	192.168.10.254	[TCP Previous segment not captured] , Ignored Unknown Record
1003	0.000000	149.50.211.10	192.168.10.254	[TCP Out-Of-Order] 443 → 56616 [ACK] Seq=5601 Ack=5652 Win=9056 Len=1400 [TCP segment of a reassembled PDU]
1005	0.000032	192.168.10.254	149.50.211.10	[TCP Dup ACK 2#2] 56616 → 443 [ACK] Seq=5652 Ack=1 Win=514 Len=0 SLE=1201 SRE=1261
1006	0.000014	192.168.10.254	149.50.211.10	[TCP Dup ACK 2#3] 56616 → 443 [ACK] Seq=5652 Ack=1 Win=514 Len=0 SLE=2801 SRE=7001 SLE=11201 SRE=5601
1007	0.000010	192.168.10.254	149.50.211.10	[TCP Dup ACK 2#4] 56616 → 443 [ACK] Seq=5652 Ack=1 Win=514 Len=0 SLE=1201 SRE=13239 SLE=2801 SRE=7001
1008	0.000714	149.50.211.10	192.168.10.254	[TCP Fast Retransmission] 443 → 56616 [ACK] Seq=1 Ack=5652 Win=9056 Len=1400 [TCP segment of a reassembled PDU]
1039	0.000000	149.50.211.10	192.168.10.254	[TCP Retransmission] 443 → 56616 [ACK] Seq=1401 Ack=5652 Win=9056 Len=1400 [TCP segment of a reassembled PDU]
1043	0.000437	149.50.211.10	192.168.10.254	[TCP Retransmission] 443 → 56616 [ACK] Seq=8401 Ack=5652 Win=9056 Len=1400
1044	0.000012	192.168.10.254	149.50.211.10	[TCP Dup ACK 1041#1] 56616 → 443 [ACK] Seq=5652 Ack=7801 Win=514 Len=0 SLE=8401 SRE=9801 SLE=11201 SRE=13239
1045	0.000457	149.50.211.10	192.168.10.254	[TCP Retransmission] 443 → 56616 [ACK] Seq=7801 Ack=5652 Win=9056 Len=1400
1046	0.000000	149.50.211.10	192.168.10.254	[TCP Retransmission] 443 → 56616 [ACK] Seq=9801 Ack=5652 Win=9056 Len=1400
1052	0.000000	149.50.211.10	192.168.10.254	[TCP Spurious Retransmission] 443 → 56616 [ACK] Seq=1 Ack=5652 Win=9056 Len=1400
1053	0.000044	192.168.10.254	149.50.211.10	[TCP Dup ACK 1049#1] 56616 → 443 [ACK] Seq=5652 Ack=13239 Win=514 Len=0 SLE=1 SRE=1401
1055	0.000335	149.50.211.10	192.168.10.254	[TCP Spurious Retransmission] 443 → 56616 [ACK] Seq=7801 Ack=5652 Win=9056 Len=1400
1056	0.000000	149.50.211.10	192.168.10.254	[TCP Spurious Retransmission] 443 → 56616 [ACK] Seq=1401 Ack=5652 Win=9056 Len=1400 [TCP segment of a reassembled PDU]
1057	0.000000	149.50.211.10	192.168.10.254	[TCP Spurious Retransmission] 443 → 56616 [ACK] Seq=8401 Ack=5652 Win=9056 Len=1400
1058	0.000058	192.168.10.254	149.50.211.10	[TCP Dup ACK 1049#2] 56616 → 443 [ACK] Seq=5652 Ack=13239 Win=514 Len=0 SLE=7801 SRE=8401
1059	0.000011	192.168.10.254	149.50.211.10	[TCP Dup ACK 1049#3] 56616 → 443 [ACK] Seq=5652 Ack=13239 Win=514 Len=0 SLE=1401 SRE=2801
1060	0.000007	192.168.10.254	149.50.211.10	[TCP Dup ACK 1049#4] 56616 → 443 [ACK] Seq=5652 Ack=13239 Win=514 Len=0 SLE=8401 SRE=9801
1071	0.002016	149.50.211.10	192.168.10.254	[TCP Spurious Retransmission] 443 → 56616 [ACK] Seq=9801 Ack=5652 Win=9056 Len=1400
1072	0.000018	192.168.10.254	149.50.211.10	[TCP Dup ACK 1049#5] 56616 → 443 [ACK] Seq=5652 Ack=13239 Win=514 Len=0 SLE=9801 SRE=11201
1074	0.000000	149.50.211.10	192.168.10.254	[TCP Dup ACK 1073#1] 443 → 56615 [ACK] Seq=127849 Ack=20501 Win=16125 Len=0
1093	0.000000	149.50.211.10	192.168.10.254	[TCP Out-Of-Order] 443 → 56610 [ACK] Seq=4294965897 Ack=1 Win=6028 Len=1400 [TCP segment of a reassembled PDU]
1094	0.000000	149.50.211.10	192.168.10.254	[TCP Out-Of-Order] 443 → 56610 [ACK] Seq=4294964497 Ack=1 Win=6028 Len=1400 [TCP segment of a reassembled PDU]
1096	0.000017	192.168.10.254	149.50.211.10	[TCP Dup ACK 1095#1] 56610 → 443 [ACK] Seq=1 Ack=4294963897 Win=588 Len=0 SLE=4294965897 SRE=1401
1097	0.000007	192.168.10.254	149.50.211.10	[TCP Dup ACK 1095#2] 56610 → 443 [ACK] Seq=1 Ack=4294963897 Win=588 Len=0 SLE=4294964497 SRE=1401
1104	0.000042	192.168.10.254	149.50.211.10	[TCP Dup ACK 1095#3] 56610 → 443 [ACK] Seq=1 Ack=4294963897 Win=588 Len=0 SLE=4294964497 SRE=2801
1106	0.000329	149.50.211.10	192.168.10.254	[TCP Fast Retransmission] 443 → 56610 [ACK] Seq=4294963897 Ack=1 Win=6028 Len=1400 [TCP segment of a reassembled PDU]
1107	0.000000	149.50.211.10	192.168.10.254	[TCP Out-Of-Order] 443 → 56609 [ACK] Seq=4294961697 Ack=1 Win=11952 Len=1400 [TCP segment of a reassembled PDU]
1108	0.000000	149.50.211.10	192.168.10.254	[TCP Out-Of-Order] 443 → 56609 [ACK] Seq=4294961697 Ack=1 Win=11952 Len=1400 [TCP segment of a reassembled PDU]

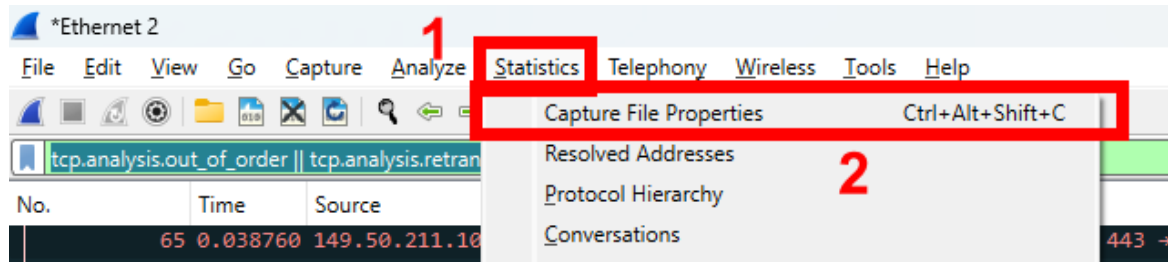
Packet details for the selected packet (No. 66):

```
> Frame 65: 66 bytes on wire (528 bits), 66 bytes captured (528 b) on interface 0  
> Ethernet II, Src: Routerboardc_98:d0:79 (6c:3b:6b:98:d0:79), Dst: Internet Protocol Version 4, Src: 149.50.211.10, Dst: 192.168.10.254  
> Transmission Control Protocol, Src Port: 443, Dst Port: 56616, Seq=1, Len=0, Win=0, Len=0, SLE=1632, SRE=1987
```

Seluruh paket berwarna hitam, artinya seluruh paket yang ditampilkan merupakan paket yang tidak sampai di tujuan

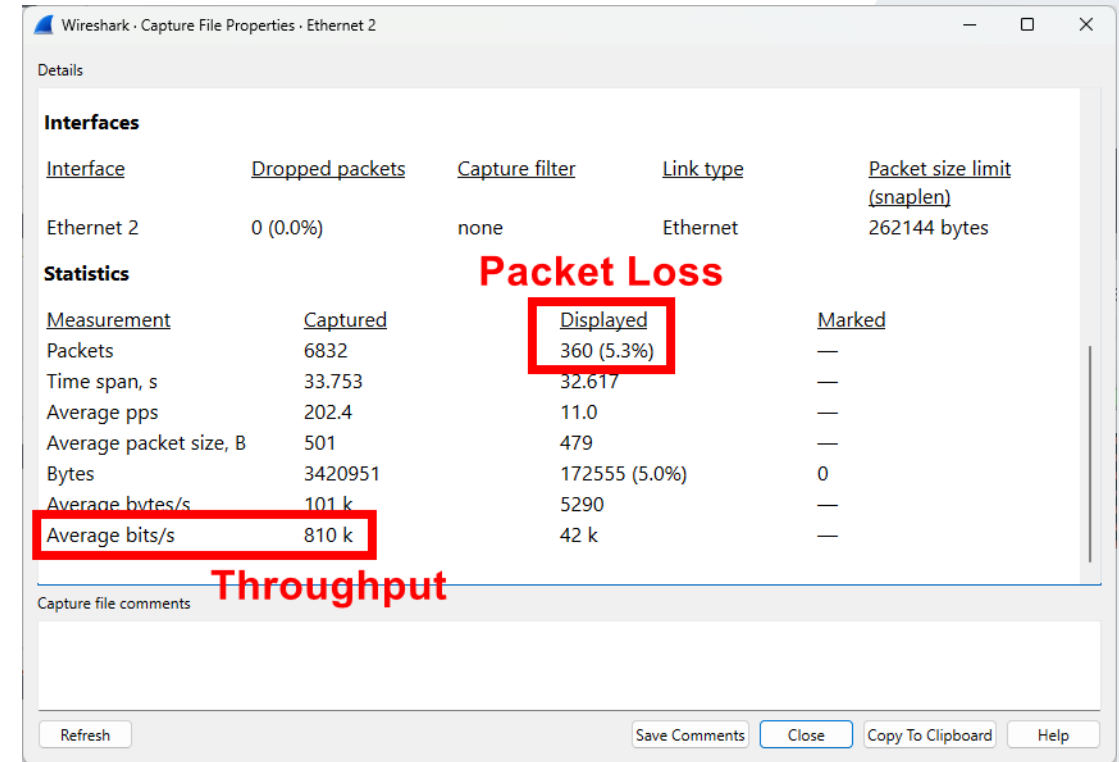
# Throughput dan Packet Loss

Untuk melihat throughput dan packet loss kita tinggal melihat statistik tampilan yang sudah difilter



Throughput didapat dari total bytes dibagi timespan dikali 8 bit per second

Packet loss didapat dari paket yang hilang dibagi jumlah paket dikali 100%



# Delay dan Jitter

Untuk menghitung delay, rubah tampilan waktu antar packet menjadi seconds since previous capture packet yang artinya waktu yang ditampilkan merupakan selisih waktu dari paket sebelumnya

The screenshot shows the Wireshark interface with the 'View' menu open. The 'Time Display Format' sub-menu is expanded, and the option 'Seconds Since Previous Captured Packet' is selected. The packet list shows several 'TCP Dup ACK' packets, indicating a network issue. The selected option is highlighted with a red box, and a red number '3' is placed next to it. A red number '2' is placed next to the 'View' menu.

Hapus filter yang digunakan agar seluruh paket ditampilkan, lalu copy paste ke microsoft excel untuk dianalisis

The screenshot shows the Wireshark interface with the packet list and packet details. The packet list shows several 'TCP Dup ACK' packets, indicating a network issue. The packet details show the structure of the captured packets.

# Delay dan Jitter

Buka excel dan paste hasil copy dari wireshark, lalu hapus semua kolom kecuali kolom waktu

	A	B	C	D	E	F	G	H	I	J	K	L
1	No.	Time	Source	Destination	Protocol							
2	1	0	149.50.211.10	192.168.10.254	Application Data							
3	2	0.001261	192.168.10.254	149.50.211.10	Application Data							
4	3	0.0001	192.168.10.254	149.50.211.10	Application Data							
5	4	0.029243	149.50.211.10	192.168.10.254	443 → 566:6 [ACK] Seq=1 Ack=234 Win=8886 Len=0							
6	5	0.000904	149.50.211.10	192.168.10.254	443 → 566:6 [ACK] Seq=1 Ack=589 Win=8898 Len=0							
7	6	0.018059	192.168.10.254	149.50.211.10	56615 → 443 [ACK] Seq=1 Ack=321 Win=514 Len=0							
8	7	0.057031	192.168.10.254	192.168.10.255	00:e0:4c:36:2b:ca > 6c:3b:6b:98:d0:79 Direction: Client->Server Type: Data							
9	8	0.000982	0.0.0.0	255.255.255.255	6c:3b:6b:98:d0:79 > 00:e0:4c:36:2b:ca Direction: Server->Client Type: Acknowledge							
10	9	0.013863	0.0.0.0	255.255.255.255	6c:3b:6b:98:d0:79 > 00:e0:4c:36:2b:ca Direction: Server->Client Type: Data							
11	10	0	0.0.0.0	255.255.255.255	6c:3b:6b:98:d0:79 > 00:e0:4c:36:2b:ca Direction: Server->Client Type: Data							



A	B	C	D	E	F	G
	0.0000000					
	0.0012610					
	0.0001000					
	0.0292430					
	0.0009040					
	0.0180590					
	0.0570310					
	0.0009820					

Siapkan bagian / cell untuk tempat hasil analisis delay dan jitter

A	B	C	D	E	F	G
	0.0000000					
	0.0012610			Delay		ms
	0.0001000			Jitter		ms
	0.0292430					
	0.0009040					
	0.0180590					

# Delay dan Jitter

Dimulai dari delay, pakai rumus  $AVERAGE(B:B)*1000$  untuk menghitung nilai rata-rata delay yang pada kolom B dan dikali dengan 1000 untuk mengubah dari second (s) menjadi milisecond (ms)

A	B	C	D	E	F	G
	0.0000000					
	0.0012610			Delay	=AVERAGE(B:B)*1000	
	0.0001000			Jitter		ms
	0.0292430					



A	B	C	D	E	F	G
	0.0000000					
	0.0012610			Delay	4.940494 ms	
	0.0001000			Jitter		ms
	0.0292430					

# Delay dan Jitter

Untuk jitter hitung selisih waktu delay antar paket dengan menggunakan rumus  $ABS(B2-B1)$  untuk setiap baris nilai delay dimulai dari delay kedua sampai terakhir

A	B	C	D	E	F	G
	0.0000000					
	0.0012610	=ABS(B2-B1)		Delay	4.940494	ms
	0.0001000			Jitter		ms
	0.0292430					



A	B	C	D	E	F	G
	0.0000000					
	0.0012610	0.001261		Delay	4.940494	ms
	0.0001000	=ABS(B3-B2)		Jitter		ms
	0.0292430	ABS(number) 143				
	0.0009040	0.028339				

# Delay dan Jitter

Lalu cari rata-rata nilai jitter pada kolom C dengan menggunakan rumus  $AVERAGE(C:C)*1000$ , dikali dengan seribu untuk mengubah second (s) menjadi milisecond (ms)

A	B	C	D	E	F	G
	0.0000000					
	0.0012610	0.001261		Delay	4.940494 ms	
	0.0001000	0.001161		Jitter	=AVERAGE(C:C)*1000	
	0.0292430	0.029143				



A	B	C	D	E	F	G
	0.0000000					
	0.0012610	0.001261		Delay	4.940494 ms	
	0.0001000	0.001161		Jitter	7.922978 ms	
	0.0292430	0.029143				
	0.0009040	0.028339				

# Delay dan Jitter

Setiap nilai tersebut kemudian dipasangkan dengan tabel standar TIPHON untuk melihat kualitas jaringan (QoS) yang dimiliki

	Throughput	Packet Loss	Delay	Jitter
Nilai	810 kbps	5.3%	4.9 ms	7.9 ms
Kualitas	Fair	Good	Sangat Bagus	Good

Standar Throughput

Kategori Throughput	Throughput	Indeks
<i>Bad</i>	0 – 338 kbps	0
<i>Poor</i>	338 – 700 kbps	1
<i>Fair</i>	700 – 1200 kbps	2
<i>Good</i>	1200 kbps – 2,1 Mbps	3
<i>Excelent</i>	>2,1 Mbps	4

Standar Packet Loss

Kategori Packet Loss	Packet Loss	Indeks
<i>Poor</i>	>25%	1
<i>Medium</i>	12 – 24%	2
<i>Good</i>	3 – 14%	3
<i>Perfect</i>	0 – 2%	4

Standar Jitter

Kategori Jitter	Jitter	Indeks
<i>Poor</i>	125 – 225 ms	1
<i>Medium</i>	75 – 125 ms	2
<i>Good</i>	0 – 75 ms	3
<i>Perfect</i>	0 ms	4

Standar Delay

Delay (ms)	Indeks	Kategori Delay
<150	4	Sangat Bagus
150 s/d 300	3	Bagus
300 s/d 450	2	Sedang
>450	1	Jelek

04

# Kesimpulan

# Kesimpulan

- Metode Quality of Service digunakan untuk memprioritaskan transmisi terhadap paket yang berasal dari aplikasi yang dirasa penting bagi organisasi/perorangan
- Terdapat bermacam-macam algoritma yang dapat dipilih dengan menyesuaikan kondisi peralatan dan kasus yang dihadapi
- Parameter QoS untuk mengukur kualitas jaringan yang dimiliki adalah throughput, Packet Loss, Delay dan Jitter
- Analisis dilakukan dengan meng-capture lalu lintas kemudian menganalisisnya menggunakan tools atau rumus yang tersedia
- Hasil analisis kemudian dibandingkan dengan standar yang berlaku

**Week 14**

---

# Network Monitoring

---