ROUTING

Telkom University

Budhi Irawan, S.Si, M.T

PENDAHULUAN

- **Routing** adalah mekanisme yang dilaksanakan pada perangkat router dijaringan (yang bekerja pada lapis 3 network) untuk mencari dan menentukan jalur yang akan dilewati sebuah paket.
- Mekanisme *pemilihan jalur* dijalankan oleh ***protokol routing*** yang akan menentukan *jalur terbaik* dengan algoritma yang berjalan didalamnya serta menggunakan parameter-parameter yang biasa disebut ***metric***.

PENDAHULUAN

- Masing-masing ***protokol routing*** bisa saja menggunakan ***algoritma*** atau ***metric*** yang berbeda-beda, sehingga walapun tujuannya sama (menentukan jalur terbaik) bisa jadi jalur yang dipilih antara ***protokol routing*** akan berbeda pula.

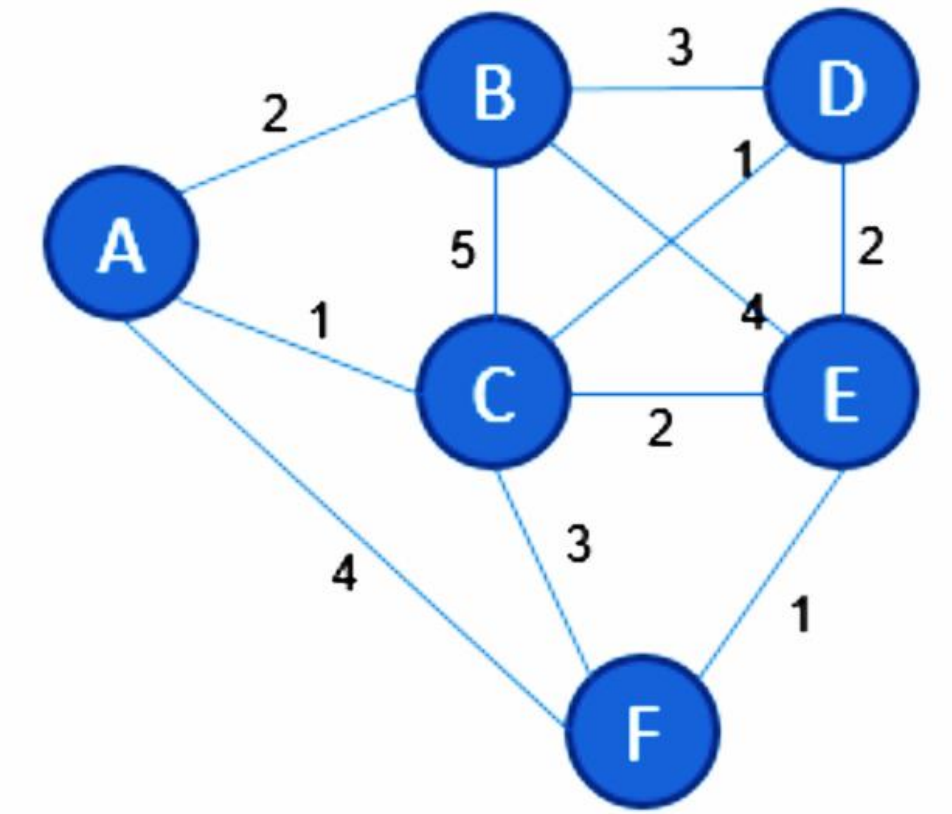
GRAPH

- Jika berbicara tentang ***routing*** maka kita tidak akan lepas dari *teori graph* sebagai dasar dari *algoritma pencarian jalur terbaik*.
- ***Graph*** sendiri adalah kumpulan dari ***titik (node)*** dan ***garis (link)*** dimana pasangan-pasangan node tersebut dihubungkan oleh ***link***.
- ***Node*** ini biasanya disebut ***simpul (verteks)*** dan segmen garis atau link disebut ***ruas (edge)***

VERTEKS DAN EDGE

- ***Verteks dan edge*** dapat diberikan tambahan informasi seperti diberi *nomor* dan dapat juga diberi *nilai* pada edge yang mana didalam jaringan komunikasi data merupakan representasi dari ***metric cost***

VERTEKS DAN EDGE



TUJUAN ROUTING

- Dalam dunia jaringan komunikasi data persoalan *pencarian jalur terbaik* dapat dimodelkan dengan ***graph***, sehingga ***graph*** dipakai untuk membantu pemecahan masalah tersebut.
- Tujuan dari ***routing*** adalah untuk mencari jalur yang paling baik, dimana paket akan dilewatkan pada jalur tersebut, sehingga paket dapat sampai ditujuan dengan cepat dan aman dari kerusakan.

METRIC

- ***Jalur terbaik*** didefinisikan sebagai pilihan jalur dengan cost terkecil.
- Dimana cost akan dibentuk atau dihitung melalui operasi matematis dari ***parameter-parameter*** atau ***metric jaringan***.
- ***Metric*** yang diamati bisa bermacam-macam dan berbeda untuk setiap ***protokol routing*** dan ***algoritma routing***.

METRIC

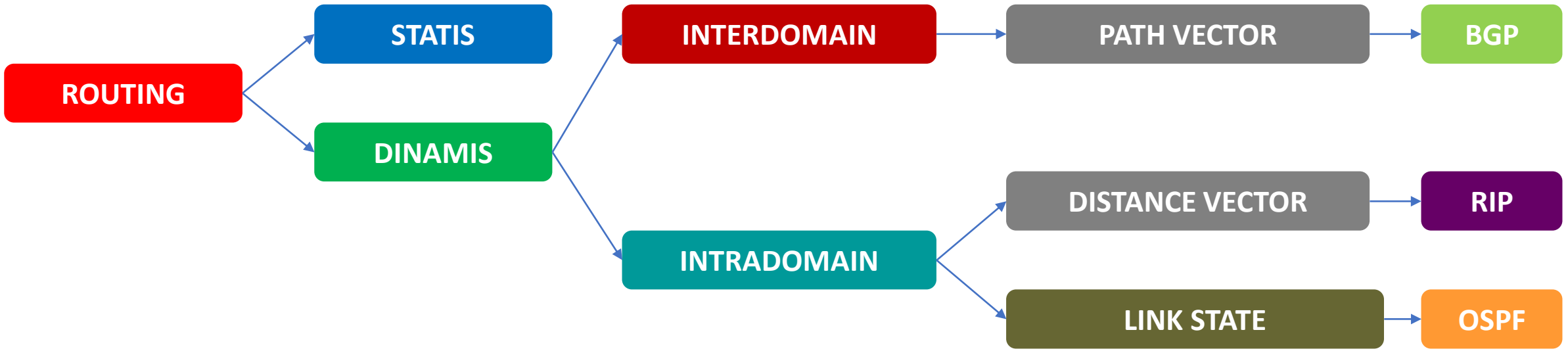
Metric yang dapat digunakan bisa saja merupakan :

- Hop Count
- Bandwidth
- Delay
- Load
- Reliability (BER)
- Maksimum Transmission Unit

METRIC

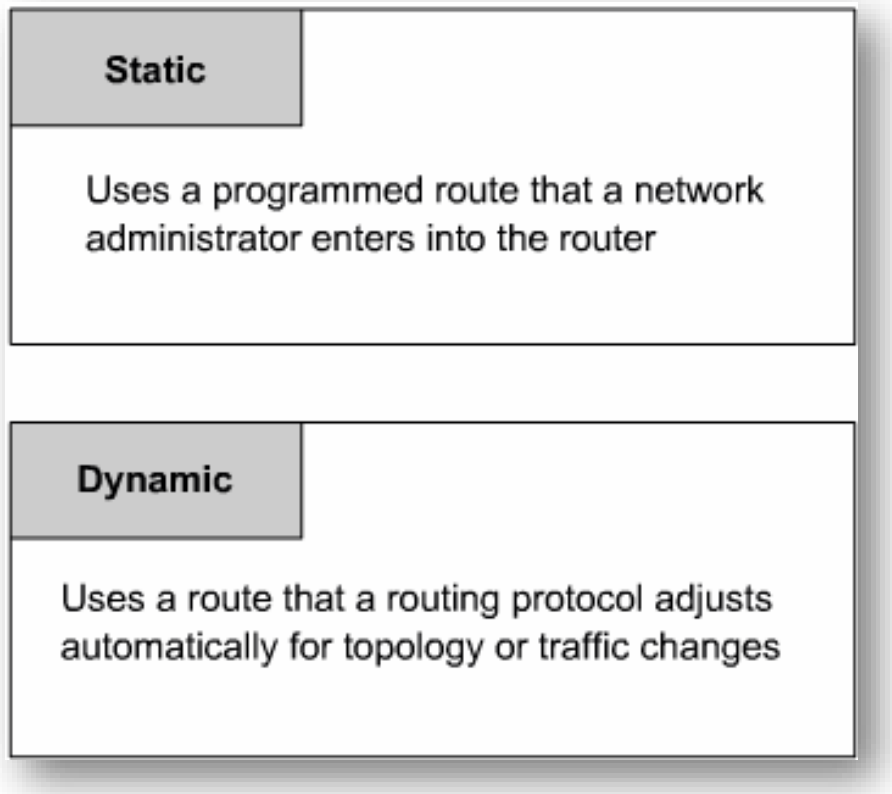
- Semakin banyak ***metric*** dan atau ***parameter*** yang digunakan maka tentu saja akan semakin dekat dengan realitas jaringan, namun ***trade off*** yang harus dibayar adalah operasi menjadi semakin kompleks sehingga berakibat beban terhadap perangkat.

KLASIFIKASI ROUTING



MEKANISME ROUTING

- Static Routing
- Dynamic Routing



ROUTING STATIS

- Mekanisme routing selain berjalan pada perangkat jaringan, dijalankan oleh ***protokol routing*** dan ***algoritma routing*** serta dapat juga dijalankan secara manual atau biasa disebut ***routing statis.***
- ***Routing statis*** dikonfigurasi secara manual oleh ***administrator*** pada perangkat jaringan, sehingga pilihan jalur paket akan selalu tetap selama konfigurasi yang ada tidak diubah oleh ***administrator***

KELEBIHAN ROUTING STATIS

- Kelebihan dari routing tipe ini adalah dalam ***segi kecepatan*** dan ***beban proses yang baik*** namun jika terjadi perubahan pada jaringan (misal : ada jaringan yang putus atau perangkat rusak) tidak dapat di antisipasi oleh perangkat secara otomatis harus menunggu ***administrator*** untuk melakukan konfigurasi ulang

KELEBIHAN ROUTING STATIS

1. Meringankan kinerja processor router
2. Tidak ada bandwidth yang diguanakn untuk pertukaran informasi dari tabel isi routing pada saat pengiriman paket
3. Routing statis lebih aman dibandingkan routing dinamis
4. Routing Statis kebal dari segala usaha hacker untuk men-spoof dengan tujuan membajak traffik

KERUGIAN ROUTING STATIS

1. Administrator jaringan harus mengetahui semua informasi dari masing-masing router yang digunakan
2. Hanya dapat digunakan untuk jaringan berskala kecil
3. Admisnistrasinya cukup rumit dibanding routing dinamis, terlebih jika banyak router yang harus dikonfigurasi secara manual
4. Rentan terhadap kesalahan saat entri data routing statis yang dilakukan secara manual.

ROUTING DINAMIS

- ***Routing dinamis*** dijalankan oleh perangkat jaringan secara **otomatis** melibatkan ***protokol routing*** dan ***algoritma routing***.
- Mekanisme routing ini dijalankan pada setiap ***node*** dan *berkolaborasi* dengan ***node*** lain didalam jaringan, sehingga dicapailah kesepekatan jalur mana yang tebaik pada pengiriman paket.

KELEBIHAN ROUTING DINAMIS

- Kelebihan dari ***mekanisme routing*** ini adalah *kemampuan pemilihan jalur yang adaptif*.
- Dapat melakukan antisipasi perubahan yang mungkin terjadi di dalam jaringan (jalur yang putus, perangkat yang rusak, trafik yang padat, dll) tetapi dibutuhkan *waktu yang lebih lama* dan *kebutuhan resource perangkat yang lebih besar* untuk menjalankan mekanisme routing ini.

KELEBIHAN ROUTING DINAMIS

1. Hanya mengenalkan alamat network yang terhubung langsung dengan routernya.
2. Tidak perlu mengetahui semua alamat network yang ada.
3. Bila terjadi penambahan suatu network baru tidak perlu semua router mengkonfigurasi. Hanya router-router yang berkaitan.
4. Lebih mudah untuk mengatur network yang besar. Akan memilih jalur lain yang ada bila suatu jalur rusak.

KERUGIAN ROUTING DINAMIS

1. Beban kerja router lebih berat karena selalu memperbarui ip table pada tiap waktu tertentu.
2. Kecepatan pengenalan network terbilang lama karena router membroadcast ke semua router hingga ada yang cocok.
3. Setelah konfigurasi harus menunggu beberapa saat agar setiap router mendapat semua Alamat IP yang ada.
4. Susah melacak permasalahan pada suatu topologi jaringan lingkup besar.
5. Update ARP table dibagikan ke semua komputer, berarti mengkonsumsi - butuh RAM untuk menentukan jalur terbaik bila terjadi down -bandwith jalur ditentukan oleh sistem, bukan admin.

PEMBAGIAN ROUTING DINAMIS

- Routing Intradomain
- Routing Interdomain

ROUTING INTRADOMAIN

- Pada ***routing intradomain*** ini dilakukan pembatasan dan pengelompokan area, karena jaringan internet yang begitu luas yang hampir tidak mungkin sebuah protokol melakukan mekanisme routing untuk seluruh jaringan internet.
- Area-area dengan mekanisme yang sejenis dan memiliki manajemen *pengaturan routing* yang sama dikelompokan menjadi sebuah area yang dinamakan ***Autonomous System***

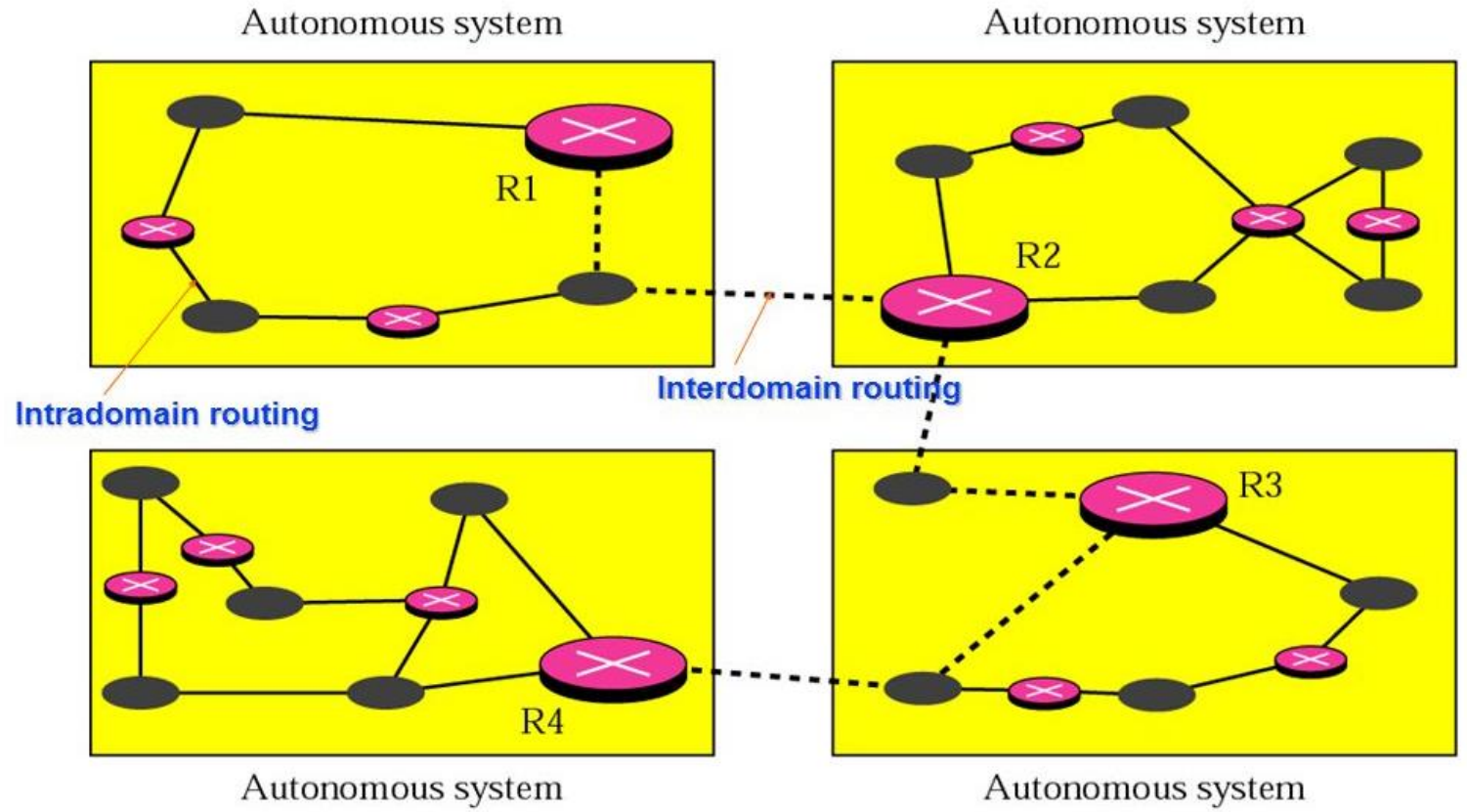
ROUTING INTRADOMAIN

- ***Protokol routing*** yang berjalan pada sebuah ***autonomous system*** disebut ***protokol intradomain***.

ROUTING INTERDOMAIN

- ***Routing Interdomain*** adalah ***mekanisme routing*** yang berjalan untuk merutekan paket yang bersumber dari satu ***atonomous system*** dengan tujuan ***autonomous system*** lain.

INTRADOMAIN VS INTERDOMAIN



ALGORITMA ROUTING

- ***Algoritma Routing*** adalah suatu mekanisme bagaimana ***protokol routing*** *mencari jalur terbaik* untuk transmisi paket.
- *Pencarian jalur terbaik* ditentukan dengan mencari total *cost/biaya yang minimum* diantara seluruh pilihan jalur yang ada di jaringan
- Cost yang dihitung berdasarkan metric yang dipergunakan pada masing-masing ***protokol routing***.

PEMBAGIAN ALGORITMA ROUTING

- Algoritma Routing Distance Vector (Bellman Ford Algorithm)
- Algoritma Routing Link State (Dijkstra Shortest Path First Algorithm)

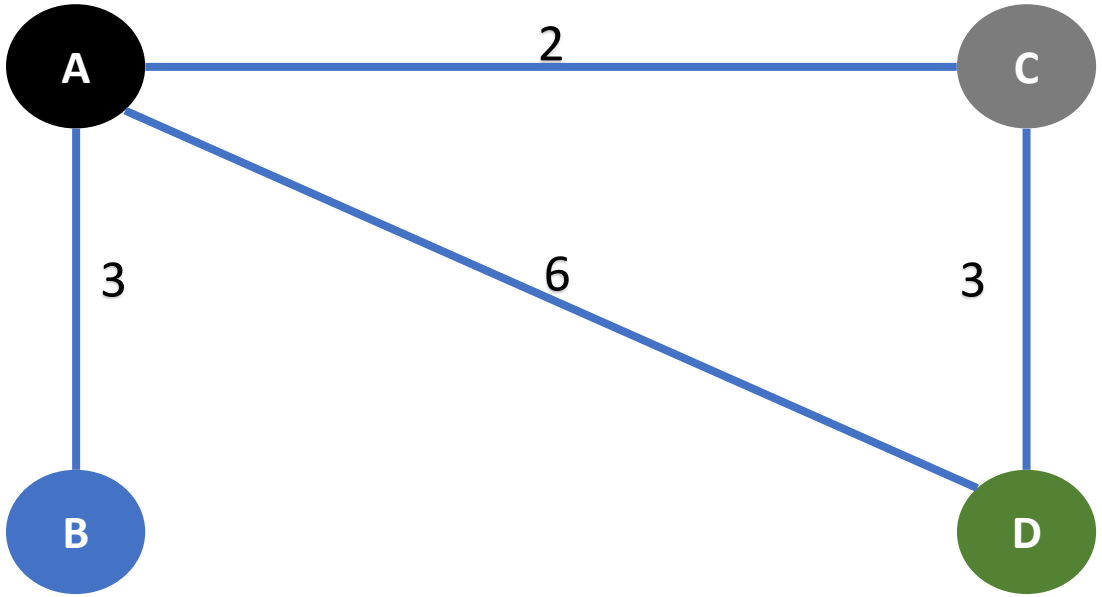
ALGORITMA DISTANCE VECTOR

- ***Algoritma Distance Vector*** ini memiliki tujuan untuk mencari *cost terkecil* dari seluruh pilihan jalur.
- Pada algoritma ini seluruh ***node*** akan memiliki dan memaintain ***vector (tabel)*** jarak minimum untuk setiap ***node*** di dalam jaringan, sehingga algoritma ini dinamakan sebagai ***Distance Vector***.

ALGORITMA DISTANCE VECTOR

To	Cost	Next
A	0	-
B	3	-
C	2	-
D	5	C

To	Cost	Next
A	3	-
B	0	-
C	5	A
D	8	A



To	Cost	Next
A	2	-
B	5	A
C	0	-
D	3	-

To	Cost	Next
A	5	C
B	8	C
C	3	-
D	0	-

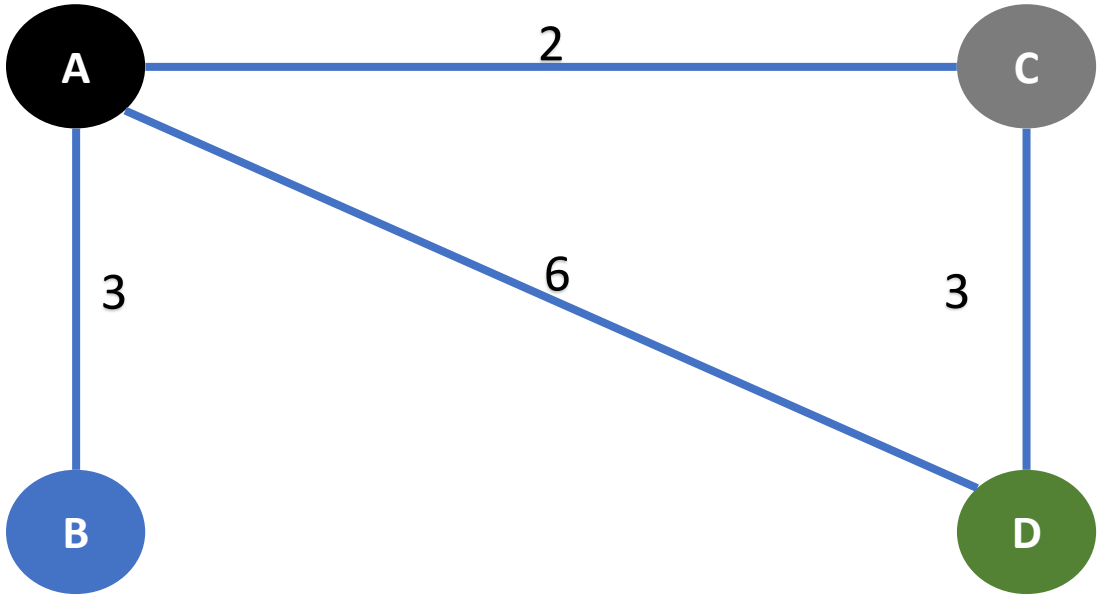
FASE INISIAL DISTANCE VECTOR

Fase Inisial

- Pada ***fase inisial***, saat pertama setiap ***node*** terhubung, maka ***tabel routing*** masing-masing ***node*** hanya dapat mendeteksi ***cost*** pada ***link*** dan ***node*** dengan koneksi langsung pada ***node*** tersebut (***immediate node***).

FASE INISIAL DISTANCE VECTOR

To	Cost	Next
A	0	-
B	3	-
C	2	-
D	6	-



To	Cost	Next
A	2	-
B	∞	-
C	0	-
D	3	-

To	Cost	Next
A	3	-
B	0	-
C	∞	-
D	∞	-

To	Cost	Next
A	6	-
B	∞	-
C	3	-
D	0	-

FASE INISIAL DISTANCE VECTOR

- Pada ***fase inisial***, tiap ***node*** hanya bisa dapat memetakan jalur-jalur dan ***node*** yang langsung ***terhubung/immediate node***
- Sebagai contoh pada ***node C*** dan ***node D*** bukan merupakan ***immediate node*** dari ***B*** sehingga ***nilai cost*** nya tidak diketahui oleh ***node B*** dan mendapatkan ***nilai infinite***.

FASE SHARING DISTANCE VECTOR

Fase Sharing

- *Tabel Routin*g yang ***konvergen*** dibangun dengan mempertukarkan ***tabel routing*** kepada ***immediate node***.
- ***Node A*** akan mengirimkan ***tabel routing*** nya kepada ***node B, C*** dan *D*.
- ***Node B*** akan mengirimkan ***tabel routingnya*** hanya ke ***node A***, karena *C* dan *D* bukan merupakan ***immediate node*** dari ***node*** B

FASE SHARING DISTANCE VECTOR

- Dari pendistribusian ***tabel routing*** inilah ***node*** menyusun peta jalur terpendek untuk seluruh tujuan ***node*** sehingga ***tabel routing*** dikatakan ***konvergen***.

FASE UPDATE DISTANCE VECTOR

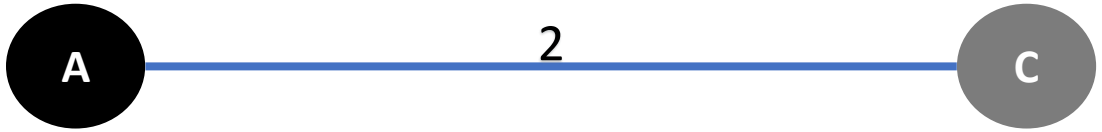
Fase Update

- ***Node*** akan mengupdate ***tabel routingnya*** ketika mendapat informasi ***tabel routing*** dari ***immediate node***.
- Melakukan perbandingan dan mengupdate ***tabel routingnya*** sesuai informasi dari ***immediate node*** dengan sudut pandang dari ***node*** asal

FASE UPDATE DISTANCE VECTOR

To	Cost	Next
A	0	-
B	3	-
C	2	-
D	6	-

To	Cost
A	0
B	3
C	2
D	6



To	Cost	Next
A	2	-
B	∞	-
C	0	-
D	3	-

FASE UPDATE DISTANCE VECTOR

- Sebagai contoh pada gambar diatas, ***node A*** mengirimkan ***tabel routing*** ke ***node C***.
- Informasi ***next node*** tidak penting bagi ***node C*** sehingga tidak akan dikirim.
- Informasi ini tidak penting karena jika harga cost ***pada tabel routing node A*** lebih kecil dari tabel di ***node C*** maka ***node C*** akan mengganti ***nilai cost*** sesuai yang terkecil dijumlahkan dengan cost menuju ***A***, dan ***ne***xt ***node*** adalah ***node A***

FASE UPDATE DISTANCE VECTOR

To	Cost
A	0
B	3
C	2
D	6

DIBANDINGKAN

To	Cost	Next
A	2	-
B	∞	-
C	0	-
D	3	-

To	Cost	Next
A	2	-
B	5	A
C	0	-
D	3	-

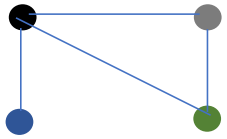
FASE UPDATE DISTANCE VECTOR

- Update akan dilakukan pada interval tertentu dan pada saat terjadi perubahan di ***node*** tergantung skema masing-masing ***protokol routing***.
- Contoh ***protokol routing*** yang menggunakan ***Distance Vector*** adalah ***RIP (Routing Information Protocol)***

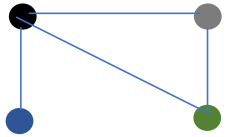
LINK STATE

- Berbeda dari ***Distance Vector*** yang ***tabel routingnya*** dibuat segera pada ***fase initial*** dan menuju ***konvergen*** dengan mempertukarkan ***tabel routing*** dengan ***immediate node***, maka pada ***link state tabel routing*** baru dibangun setelah seluruh node dijaringan terkumpul.
- Secara praktis seluruh ***node*** memiliki pandangan yang utuh terhadap peta jaringan, yang kemudian akan dicari jalur terpendeknya untuk menyusun ***tabel routing***

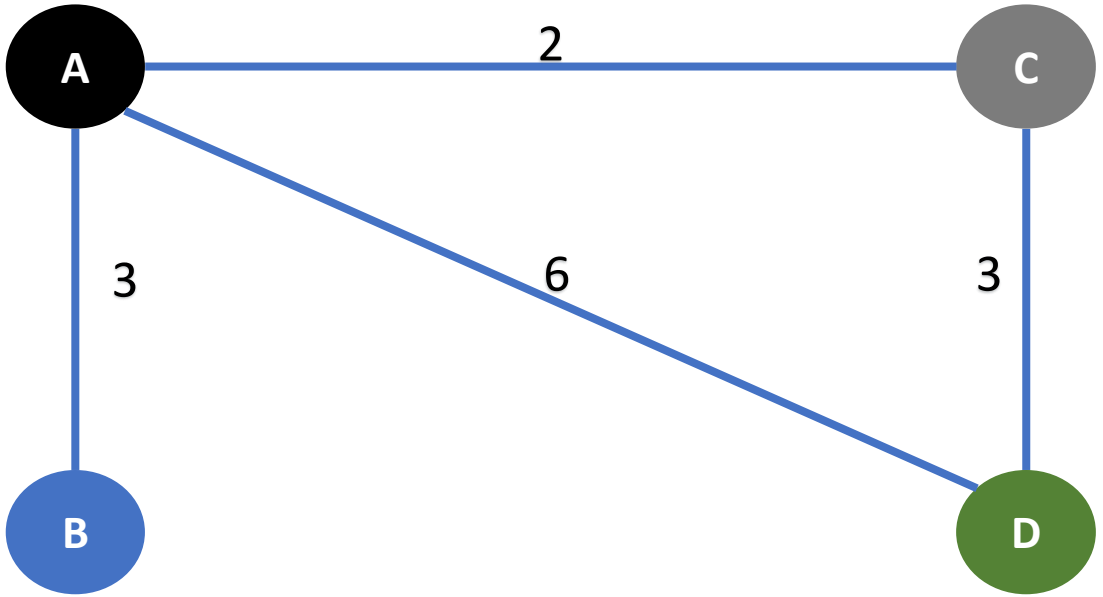
TABEL ROUTING LINK STATE



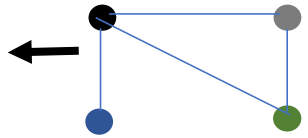
To	Cost	Next
A		
B		
C		
D		



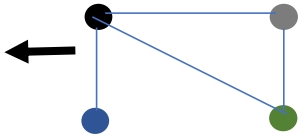
To	Cost	Next
A		
B		
C		
D		



To	Cost	Next
A		
B		
C		
D		



To	Cost	Next
A		
B		
C		
D		



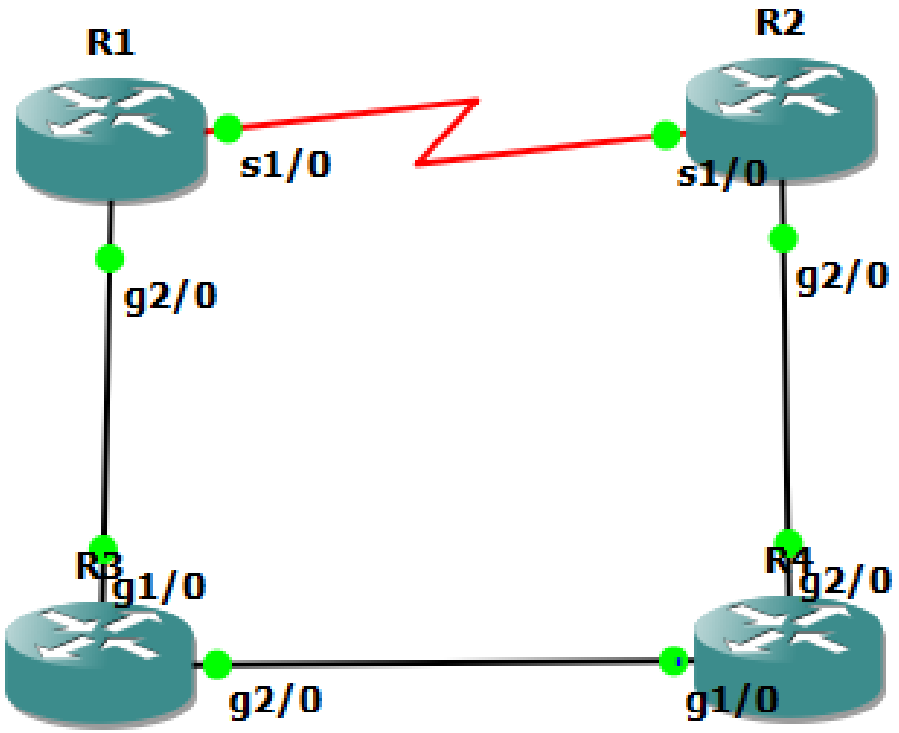
TABEL ROUTING LINK STATE

- Dalam penyusunan tabel routingnya secara umum link state memiliki 4 tahap :

1. Membuat status setiap link pada masing-masing node, dan informasinya dimasukan kedalam LSP (Link State Protocol)
2. Mengirimkan LSP keseluruh node yang ada di jaringan
3. Menyusun link status menjadi topologi dimasing-masing node
4. Melakukan kalkulasi jalur terpendek guna menyusun tabel routing

- ***Protokol Routing*** yang menggunakan ***algoritma link state*** adalah ***OSPF (Open Shortest Path First)***

LINK STATE VS DISTANCE VECTOR



1. Kalau menggunakan metode **Distance Vector** maka dari R1 ke R2 akan lewat mana?
2. Kalau menggunakan metode **Link State maka** dari R1 ke R2 akan lewat mana?

Jawaban

1. **Distance Vector** akan menggunakan jalur **R1 langsung ke R2** karena dianggap paling dekat walaupun koneksi menggunakan serial interface yang kecepatannya jauh lebih lambat dari gigabyte port.
2. **Link State** akan menggukanan jalur **R1-R3-R4-R2**, karena walaupun hop count nya jauh lebih banyak tapi bandwitdhnya lebih besar/ lebih cepat