



Jaringan Komunikasi Data E-Learning

Presents:

Data and Logical Link Control (LLC)




Minggu 4





Outline

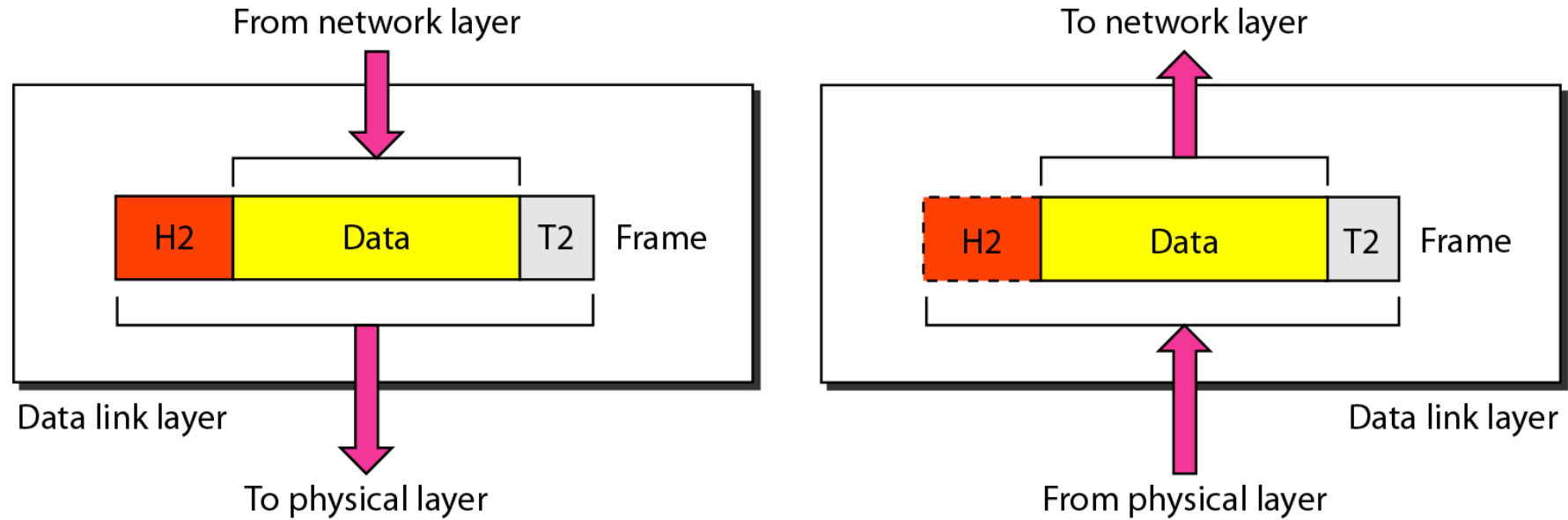
- Data link layer
 - Error Detection (Parity, CRC, Checksum)
 - Error Correction (Block Parity, Hamming) and FEC
 - Simple Protocol (Stop and Wait, Piggybacking)
- 



Data Link Layer



Data Link Layer



Lapisan data link bertanggung jawab untuk memindahkan frame dari satu hop (node) ke yang berikutnya.

Link

- Link → Jalur yang menghubungkan antar 2 elemen jaringan (node-node atau terminal-node)
- Kumpulan link (+ node-node) = jaringan
- Fungsi link sangat vital, maka OSI menetapkan protokol lapis 2 (datalink)
- Datalink = mengatur agar komunikasi di link tersebut berjalan **benar** dan **lancar**
- Tidak ada keharusan jenis link dalam jaringan sama, boleh memilih teknologi link (fisik maupun protokol) untuk setiap link




Terdapat 2 macam link :

- 1) link fisik dan
 - 2) link logik (contoh: virtual path yang terdiri atas virtual channel)
- 




Tugas Datalink

- Pembukaan hubungan dan penutupan hubungan
 - Melakukan pengaturan akses media → Media Access Control
 - Melakukan kendali atas kesalahan yang mungkin terjadi
tools: pariti, CRC, dll
 - Melakukan pengendalian banyaknya data yang dikirim untuk menghindari kemacetan (kongesti),
tools: sliding windows dll
- 

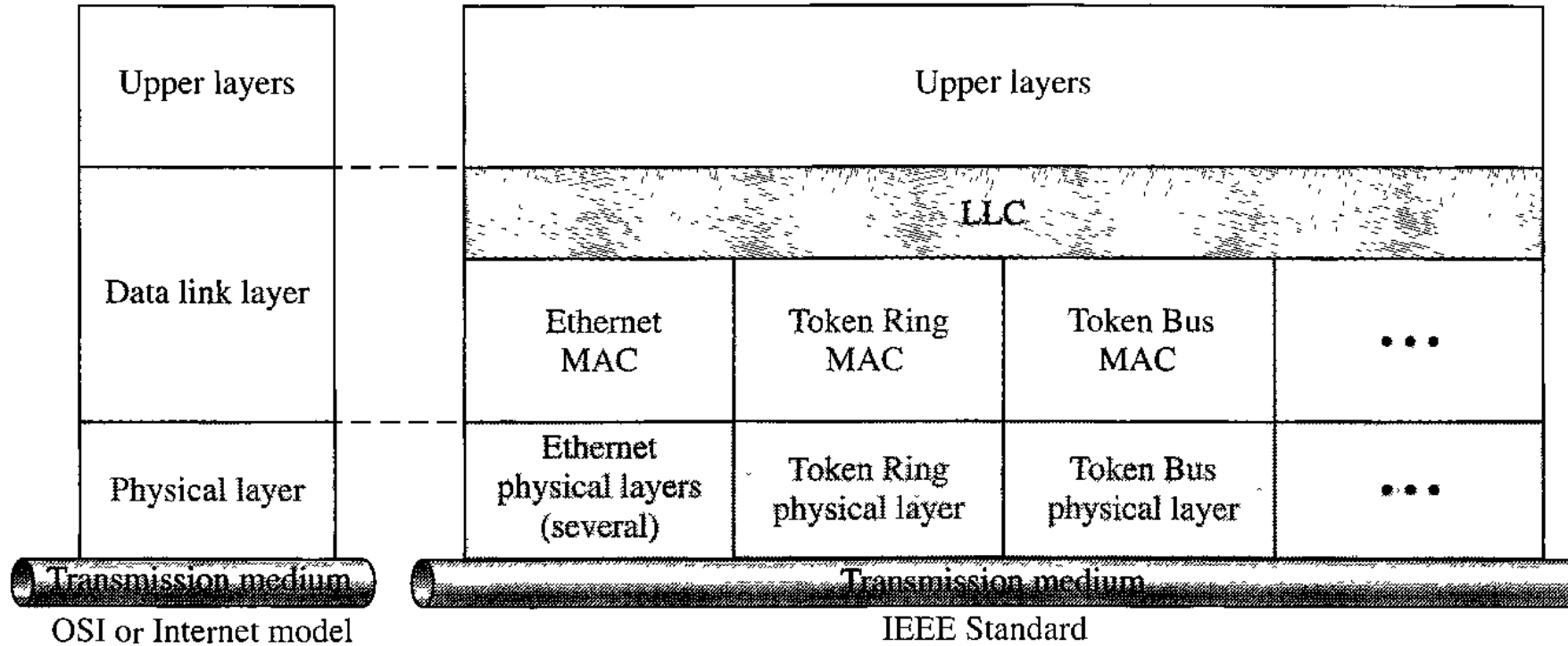


Proses Hubungan di Link

- Dua jenis proses hubungan di link
 - Memerlukan connection setup
 - Hubungan langsung
 - Connection setup
 - Ada banyak path yang bisa dipilih
 - Untuk hubungan yang sangat handal
 - Tersedia berbagai pilihan kecepatan komunikasi
 - Hubungan langsung
 - Tanpa pilihan jalur dan kecepatan komunikasi
 - Point-to-point connection
- 


Sub Layer pada Data Link

LLC: Logical link control
MAC: Media access control






Sub Layer pada Data Link

- Sub layer LLC (Logical Link Control) bertanggung jawab terhadap kontrol data link, termasuk flow control dan error control.
 - Sub layer MAC (Media Access Control) bertanggung jawab terhadap shared media akses, memastikan bahwa 2 device tidak bicara bersamaan.
- 




Error Detection





Metode Error Detection (Deteksi Kesalahan)

- Agar bisa melakukan kendali kesalahan, syarat mutlak yang harus ada adalah adanya mekanisme deteksi kesalahan
 - Beberapa metode yang umum digunakan:
 - Pariti → paling sederhana
 - CRC → lebih sulit, membutuhkan kemampuan komputasi
 - Checksum → operasi word
- 

Pariti

- Penambahan 1 bit sebagai bit deteksi kesalahan
- Terdapat 2 jenis pariti: genap dan ganjil
 - Pariti genap = jumlah bit 1 dalam kode adalah genap
 - Pariti genap = $d_1 \text{ xor } d_2 \text{ xor } \dots \text{ xor } d_n$
 - Pariti ganjil = jumlah bit 1 dalam kode adalah ganjil
 - Pariti ganjil = $(d_1 \text{ xor } d_2 \text{ xor } \dots \text{ xor } d_n) \text{ xor } 1$
- Sistem sederhana dan mudah dibuat hardwarenya (di PC digunakan IC 74LS280)

Cyclic Redundancy Check

Sisi Pengirim

Merupakan hasil operasi pembagian biner dengan suatu pembagi tertentu (generator polinomial)

- Pembagi : $D_n D_{n-1} \dots D_1$
- Deretan bit : $b_1 b_2 b_3 \dots b_m$
- Operasi :
 $(b_1 b_2 b_3 \dots b_m)_{n-1} / D_n \dots D_1 \rightarrow \text{sisanya } (R_{n-1} \dots R_1)$
- Dikirim $b_1 b_2 b_3 \dots b_m R_{n-1} \dots R_1$




Cyclic Redundancy Check

Sisi Penerima (1)

Oleh penerima dilakukan operasi yang sama

- $b_1 b_2 b_3 \dots b_m R_{n-1} \dots R_1 / D_{n-1} \dots D_1 \rightarrow \text{sisanya } (r_{n-1} \dots r_1)$
- Data benar jika $r_{n-1} \dots r_1 = 0$
- Data salah jika $r_{n-1} \dots r_1 \neq 0$

Pembagi standar internasional


- CRC-16 \rightarrow 110000000000000101
 - CRC-ITU \rightarrow 10001000000100001
 - CRC-32 \rightarrow 100000100100000010001110110110111
- 



Cyclic Redundancy Check

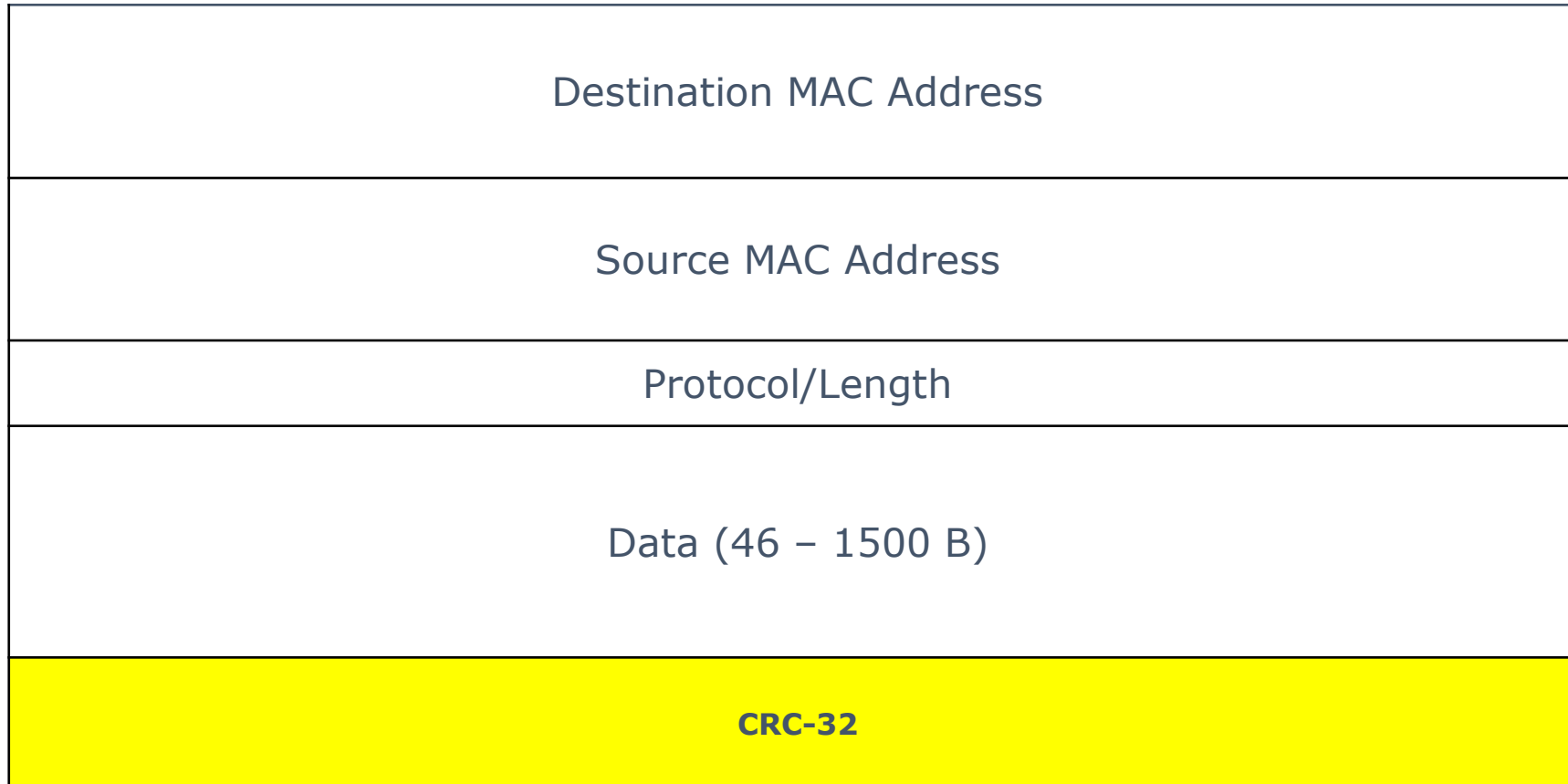
Sisi Penerima (2)

Jika diperlukan pembagi boleh tidak menggunakan standar ini asal memenuhi:

- Diawali dan diakhiri dengan bit 1 (1xxxxx1)
 - Jumlah minimum bit “1” : 3 bit
 - Agar bisa mendeteksi jumlah bit kesalahan ganjil : harus habis dibagi oleh $(11 = X + 1)$
- 

Penggunaan: Paket LAN (MAC)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16






Checksum

Pada CRC terdapat perhitungan xor sebanyak jumlah bit data, sehingga memerlukan kemampuan komputasi yang cukup besar


Diciptakan metode checksum (untuk mengurangi perhitungan) pada beberapa jenis transmisi tidak perlu kecanggihan CRC atau sudah melakukan CRC di lapis lain



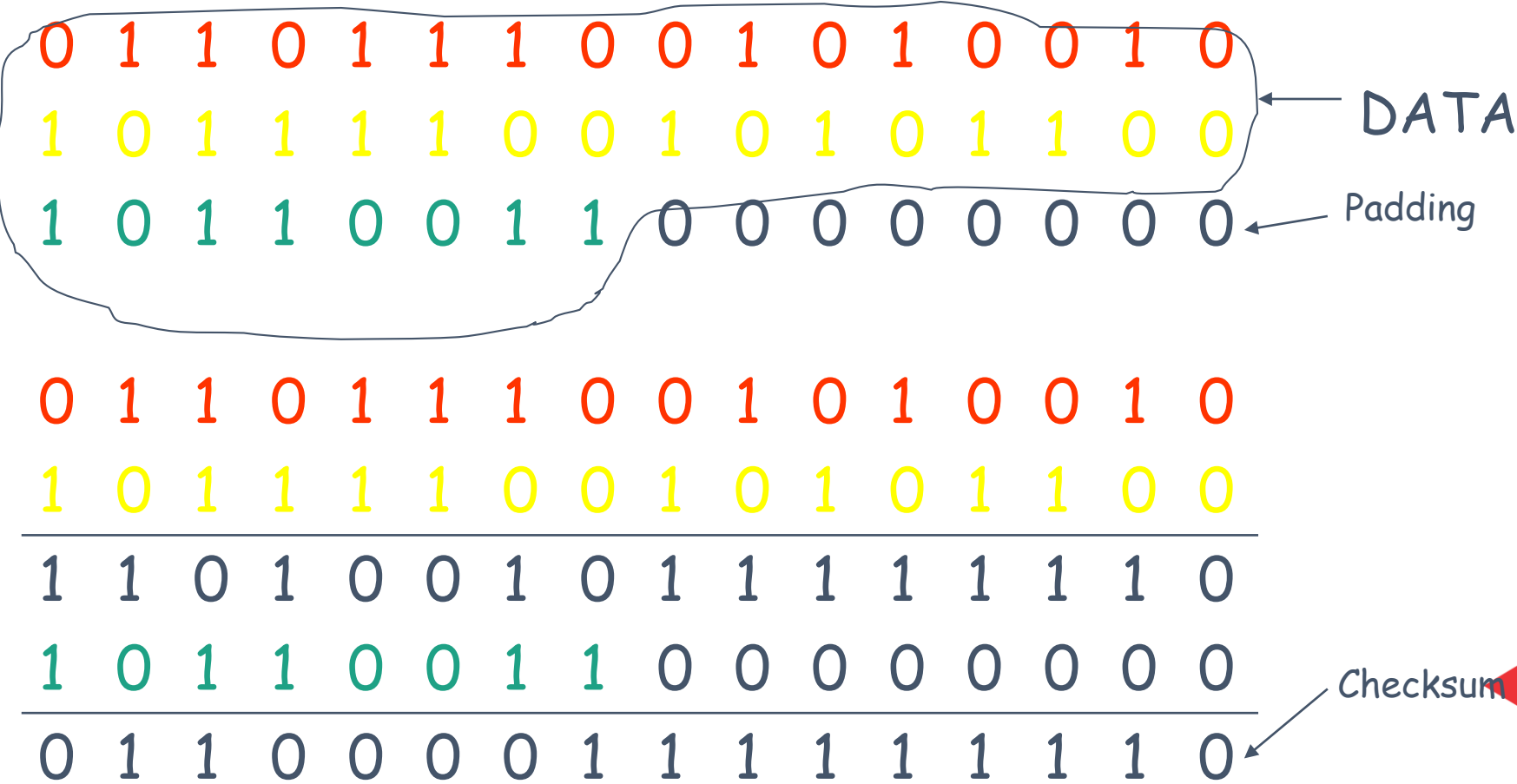


Checksum

Cara perhitungan checksum:

- 1) Data dibagi menjadi kelompok-kelompok 16 bit (word)
 - 2) Word pertama di xor dengan word kedua
 - 3) Hasil di xor dengan word ketiga, keempat, ...sampai word terakhir (jika bit-bit terakhir tidak cukup untuk menjadi word, ditambahkan padding bit '0' sampai membentuk word)
 - 4) Hasil akhir (16 bit) = checksum
- 

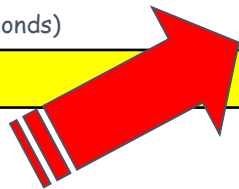
Contoh Perhitungan Checksum



Penggunaan Checksum: IP

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
 Priority (0-7) low high high ← "1"

Version		Header length (dword)		Precedence		D	T	R	unused
Total length									
<i>Identification</i>									
D	M	<i>Fragment offset</i>							
Time to live (seconds)					Protocol				
Header checksum									
Source IP address									
Destination IP address									
Option (0 word atau lebih)									
Data ≤ 64 kB									



Penggunaan Checksum: TCP

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Source port																
Destination port																
		Sequence number														
		Acknowledge number														
Header length				Reserved				URG	ACK	PSH	RST	SEQ	FIN			
Windows																
Checksum																
Urgent pointer																
Options																
								Padding								
User data																




Error Correction





Teknik Error Correction

- ❑ Backward Error Control (BEC)
 - ❑ Forward Error Control (FEC)
- 


Backward Error Control (BEC)

Kemampuan deteksi kesalahan digunakan untuk melakukan perbaikan kesalahan (error correction) dengan cara meminta pengiriman ulang jika paket yang diterima salah

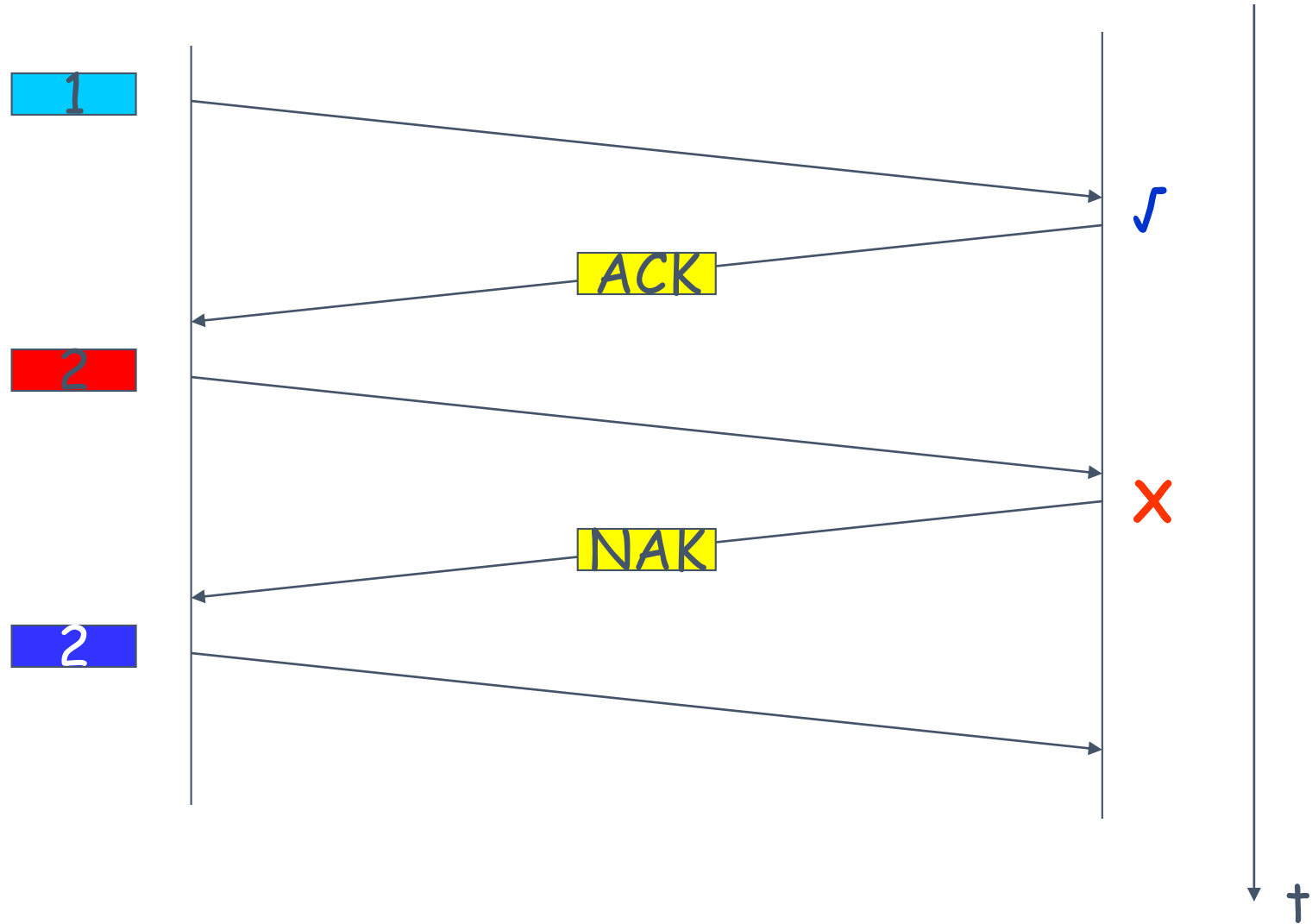




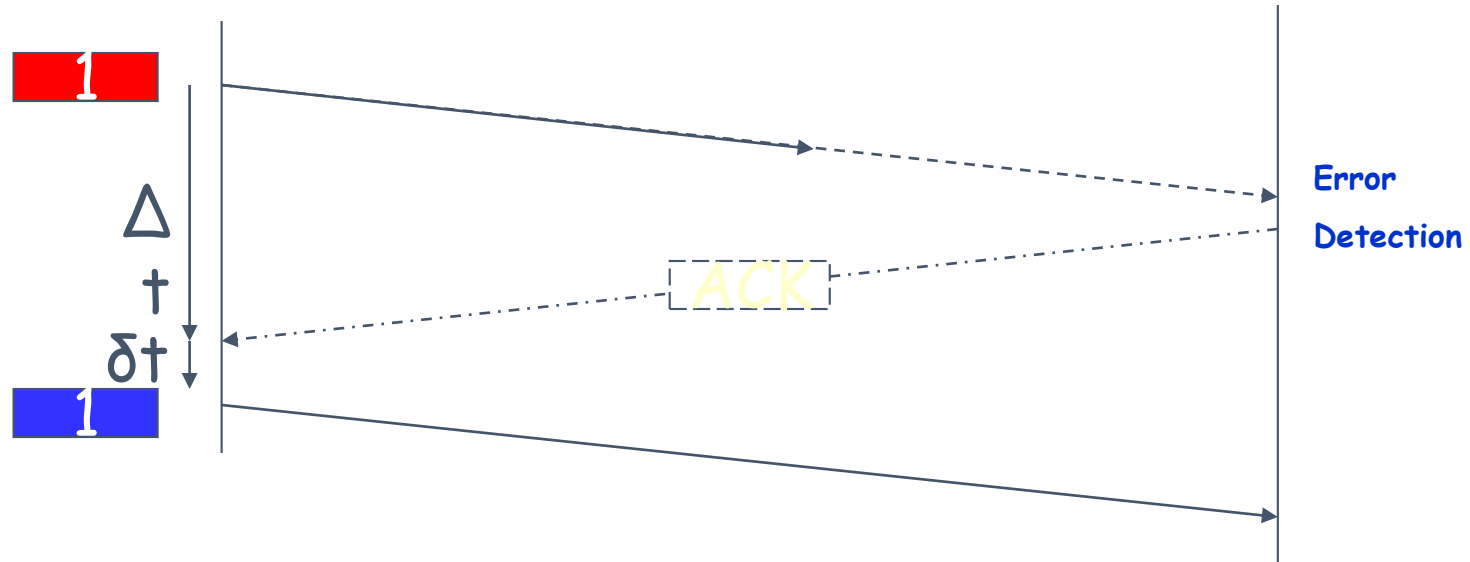
Backward Error Control: ARQ

- ARQ = Automatic Repeat reQuest
 - ARQ akan mengulang / tidak mengulang pengiriman data sesuai dengan feedback dari penerima
 - Feedback dari penerima
 - ACK = acknowledge → data diterima benar
 - NAK = not acknowledge → data diterima salah
- 

ARQ : Idle RQ



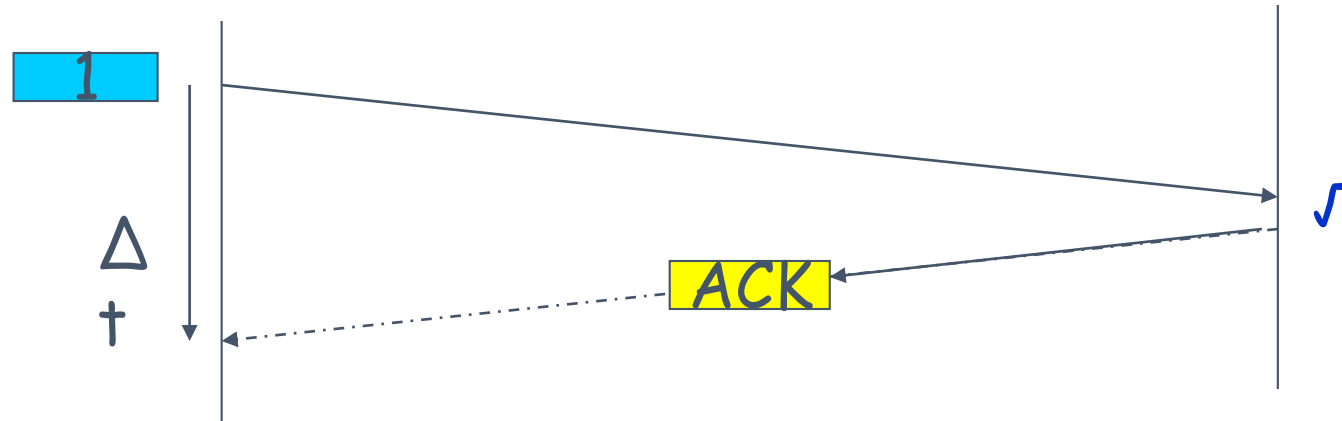
Kasus 1: jika paket tidak sampai



Pengirim menunggu feedback sampai $\Delta t + \delta t$, jika tidak ada respon maka pengirim harus mengirimkan kembali paket.

Waktu disebut dengan waktu *timeout*.

Kasus 2: feedback tidak sampai



Diperlakukan sama dengan kondisi kasus 1 (time-out)



Kapankah pengirim mengirim ulang paket?

- Jika mendapat feedback NAK
- Jika timeout
- Jika mendapat feedback yang tidak dimengerti

Kesimpulan:

pengirim mengirim ulang paket → Jika tidak mendapat
ACK



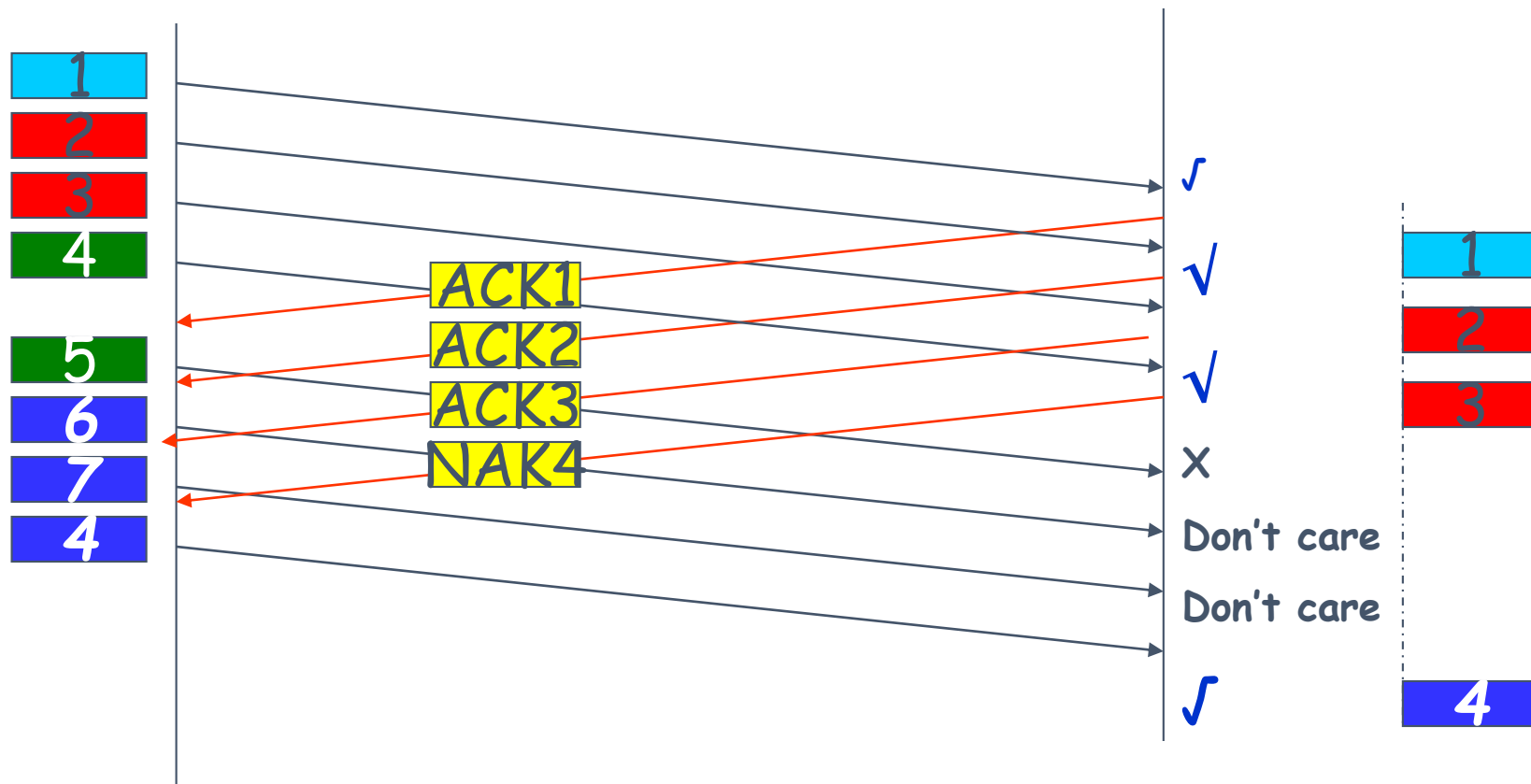


ARQ: Idle RQ

- Paket yang diterima terjaga urutannya
 - Efisiensi saluran paling rendah
- Cocok digunakan untuk saluran transmisi yang sangat jelek kualitasnya (banyak error)

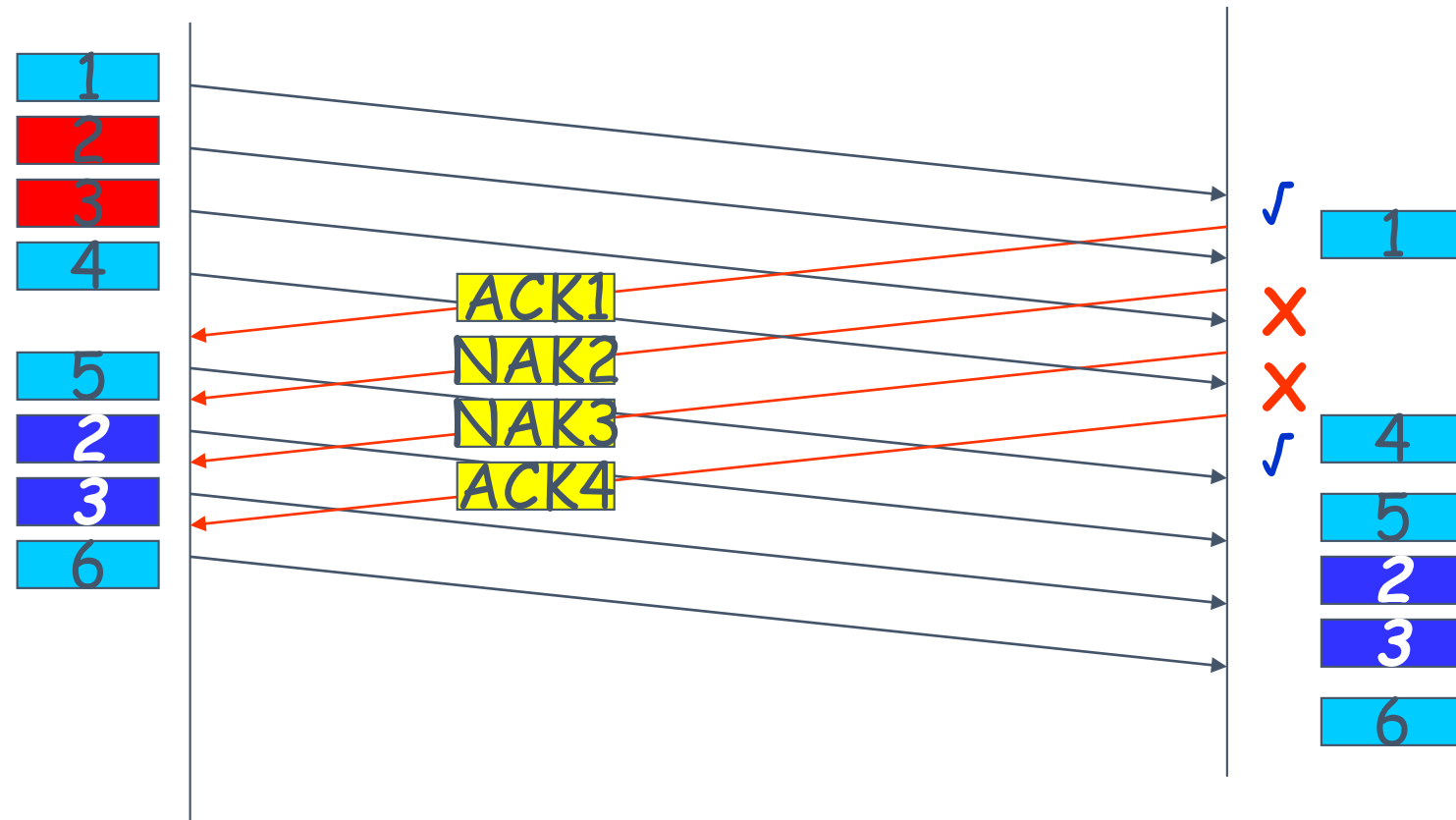


Kasus Lain Go Back N



ARQ: Selective Repeat

- Hanya mengirim ulang untuk paket yang salah
- ▶ ● Paket diterima tidak berurutan
- Efisiensi saluran tinggi (dibandingkan idle RQ)






Backward Error Control

Error control = error detection + ARQ


Kelemahan:

waktu yang diperlukan untuk mengirim dengan benar
adalah minimal 2 x waktu propagasi





Masalah Penggunaan BEC

- Backward EC menyebabkan delay pengiriman paket yang cukup besar tergantung dari berapa kali paket tersebut harus dikirim
 - Untuk sistem transmisi jarak jauh dimana delay propagasi sangat besar (kelas detik, menit atau jam) BEC tidak bisa menjadi pilihan
 - Dalam aplikasi multimedia, ketepatan waktu kedatangan lebih utama dibandingkan dengan 'kebenaran' data, BEC menyebabkan delay yang lewat batas toleransi waktu
- 




Forward Error Control (FEC)

Forward Error Correction (FEC) digunakan untuk memecahkan masalah penggunaan BEC

Prinsip dasar FEC:

Penerima mampu membetulkan sendiri kesalahan data yang sudah diterima, karena selain menerima data juga menerima bit-bit redundansi yang diperlukan





Jenis-Jenis FEC

Metode FEC yang umum dikenal:

- Block Parity
 - Hamming Code
 - Turbo Code, RS Code, BCH Code
- 



References

- *Data Communications and Networking*, 5th Edition, Behrouz A. Forouzan, McGraw Hill, 2013
 - *Data and Computer Communications*, 10th Edition, William Stallings, Pearson Education, 2014
 - *Computer Networking: A Top Down Approach*, 7th Global Edition, James F. Kurose & Keith W. Ross, Pearson Education, 2017
- 