

Data Link Control

Bagian tanggung jawab terpenting dari data link layer adalah flow control dan error control. Atau seluruh bagian ini dinamakan data link control.

Framing

Bit-bit dalam layer data link dirangkai ke dalam frame-frame.

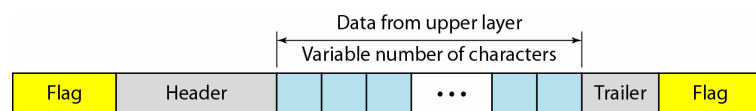
Cara memecah bit-bit menjadi frame di antaranya adalah menyisipkan gap waktu di antara dua frame. Cara lainnya adalah menggunakan karakter penghitung, karakter awal dan akhir, flag awal dan akhir, dan pelanggaran kode layer fisik.

Fixed size framing

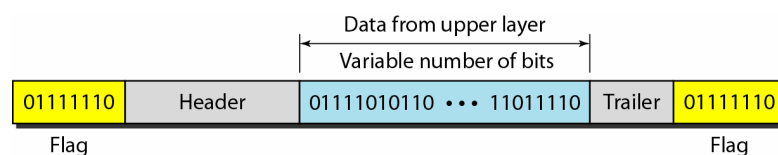
Frame dapat dibagi-bagi dalam ukuran yang sama jumlah bitnya.

Variable size framing

Frame yang jumlah bitnya berbeda-beda/ bervariasi disebut variable size framing.



Gambar 11.1 – frame dalam protokol berorientasi karakter



Gambar 11.2 – frame dalam protokol berorientasi bit

Dalam protokol berorientasi karakter, byte stuffing/ pengisian byte, contohnya adalah proses menambahkan 1 byte ekstra ketika ada flag atau karakter escape dalam teks.

Dalam protokol berorientasi bit, bit stuffing/ pengisian bit, contohnya adalah proses menambahkan satu 0 ekstra setiap lima bit 1 mengikuti sebuah 0 dalam data.

Flow dan Error Control

Flow Control

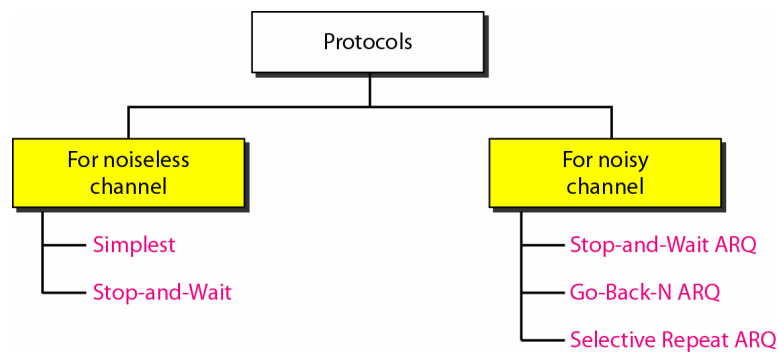
Flow control mengacu pada sekumpulan prosedur yang digunakan untuk membatasi jumlah data yang dikirim pengirim sebelum menerima acknowledge.

Error Control

Error control dalam data link layer berbasis pada pengulangan request otomatis/ automatic repeat request (ARQ), untuk retransmisi data.

Protokol

Potokol biasanya diimplementasikan dalam software menggunakan bahasa pemrograman. Dalam mengimplementasikan flow control dan error control, kita gunakan bentuk protokol.

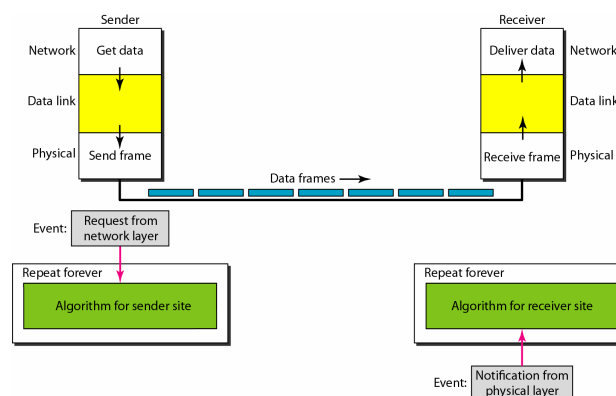


Gambar 11.3 – taksonomi protokol dalam data link control

Noiseless Channel

Simplest Protocol

Simplest protocol adalah protokol pengiriman frame tanpa flow control dan error control. Jadi pengirim hanya membentuk frame paket data upper layer dan mengirimkannya, sedangkan penerima menerima frame lalu meneruskannya ke upper layer.



Gambar 11.4 – desain simplest protocol tanpa flow control dan error control

Algoritma yang digunakan pengirim adalah :

```

1  while(true)                // ulangi selamanya
2  {
3      WaitForEvent();         // sleep sampai terjadi event
4      if(Event(RequestToSend)) // ada paket untuk dikirim
5      {
6          GetData();
7          MakeFrame();
8          SendFrame();         // kirim frame
9      }
10 }

```

Sedangkan algoritma yang digunakan penerima :

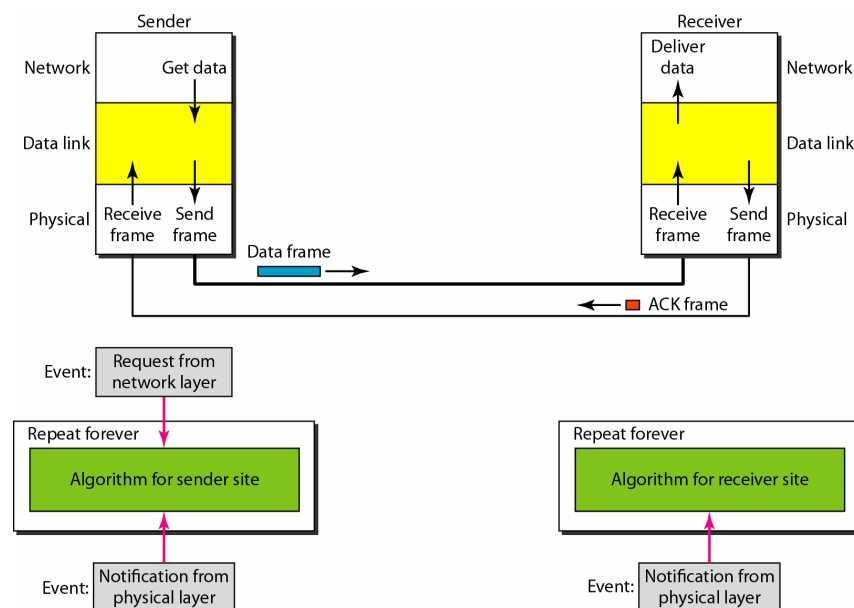
```

1  while(true)                // ulangi selamanya
2  {
3      WaitForEvent();         // sleep sampai terjadi event
4      if(Event(ArrivalNotification)) // Frame data diterima
5      {
6          ReceiveData();
7          ExtractData();
8          DeliverData();       // kirim data ke network layer
9      }
10 }

```

Stop and Wait Protocol

Protokol stop and wait adalah protokol dengan fasilitas flow control dan error control paling sederhana. Pada sisi pengirim, ia mengirimkan frame ke penerima lalu kemudian penerima mengirimkan ACK/ acknowledge ke pengirim jika frame tersebut telah diterima. Setelah ACK diterima oleh pengirim, maka frame berikutnya dikirimkan, dan seterusnya.



Gambar 11.5 – Disain protokol stop and wait

Algoritma pada pengirim adalah :

```

1  while(true)                // ulangi selamanya
2  canSend = true              // ijin frame pertama dikirim
3  {
4      WaitForEvent();         // sleep sampai terjadi event
5      if(Event(RequestToSend) AND canSend)
6      {
7          GetData();
8          MakeFrame();
9          SendFrame();         // kirim frame
10         canSend = false;     // tidak bisa kirim sampai ACK
11     }
12     WaitForEvent()          // sleep sampai terjadi event
13     if(Event(ArrivalNotification)) // ACK diterima
14     {
15         ReceiveFrame();      // frame ACK diterima
16         canSend = true;
17     }
18 }

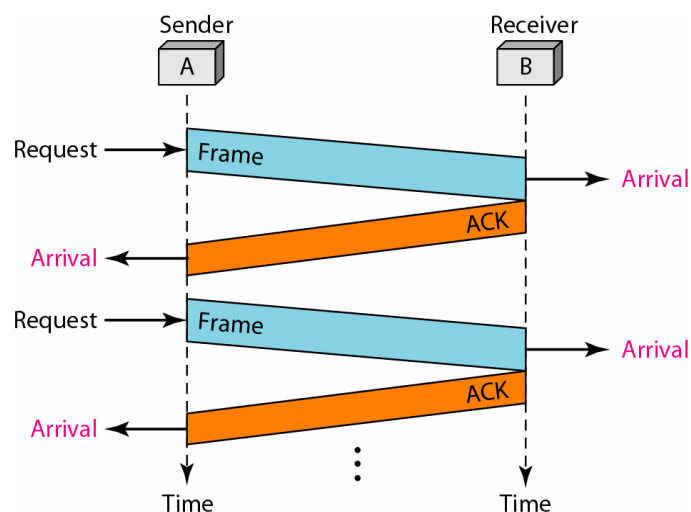
```

Algoritma pada sisi penerima :

```

1  while(true)                // ulangi selamanya
2  {
3      WaitForEvent();         // sleep sampai terjadi event
4      if(Event(ArrivalNotification)) // Frame data diterima
5      {
6          ReceiveData();
7          ExtractData();
8          DeliverData();       // kirim data ke network layer
9          SendFrame();         // kirim frame ACK
10     }
11 }

```



Gambar 11.6 – Aliran data dengan protokol stop and wait

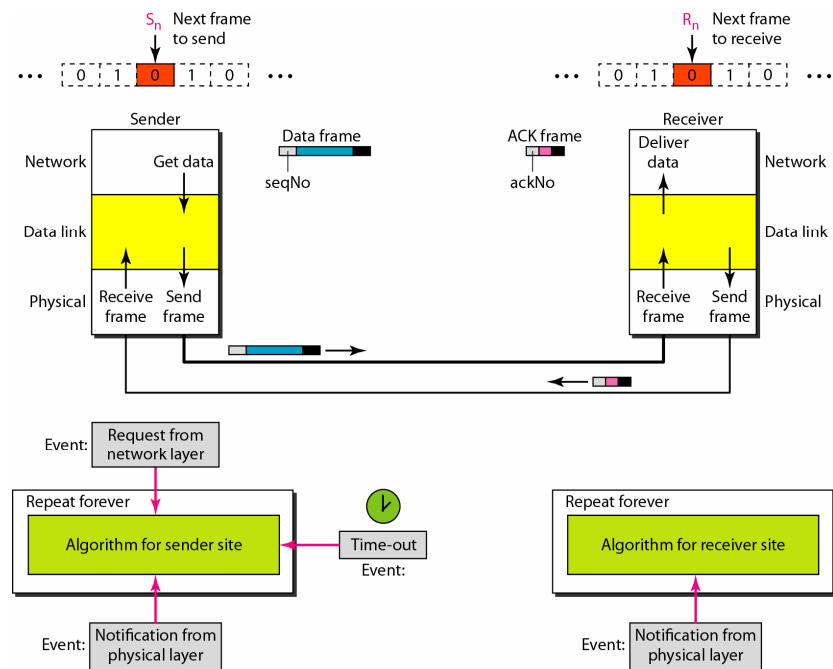
Noisy Channel

Dalam kenyataannya tidak ada kanal yang noiseless, sehingga protokol di atas tidak realistis. Yang nyata adalah kanal yang dapat diganggu oleh interferensi atau gangguan fisik lainnya. Untuk kanal jenis ini akan dibahas tiga protokol yang dapat digunakan.

Stop and Wait Automatic Repeat Request (ARQ)

Stop and Wait ARQ adalah modifikasi stop and wait protocol yang memiliki kemampuan error control dan flow control. Ia memiliki fitur :

- Perangkat pengirim tetap menyimpan salinan frame terakhir yang dikirimkan sampai ia menerima acknowledge terhadap frame tersebut. Salinan frame tersebut disimpan untuk berjaga-jaga jika frame tersebut hilang atau rusak dalam transmisi.
- Untuk identifikasi, baik frame data maupun acknowledge (ACK) diberi nomor 0 dan 1 bergantian. Frame data 0, dibalas dengan ACK 1, artinya penerima telah menerima frame data 0 dan sedang menanti frame data 1.
- Frame yang rusak atau hilang diperlakukan sama oleh penerima. Jika penerima mendeteksi error pada frame yang diterima, ia mengabaikan frame tersebut dan tidak mengirim ACK. Jika penerima menerima frame yang tidak berurut, ia tahu ada frame hilang, ia mengabaikan frame tersebut juga.
- Pengirim memiliki variabel kontrol S, yang menyimpan nomor frame yang sedang dikirim (0 dan 1). Sedangkan variabel kontrol, R yang menyimpan nomor frame yang diharapkan untuk diterima (0 dan 1).
- Pengirim memulai timer ketika ia mengirim frame. Jika ACK tidak diterima dalam waktu tertentu, pengirim mengasumsikan bahwa frame hilang atau rusak, dan ia mengirim ulang frame tersebut.
- Penerima mengirimkan hanya ACK yang positif (berhasil) terhadap frame yang berhasil diterima. Ia diam jika ada frame rusak atau hilang. Nomor ACK selalu mendefinisikan nomor frame berikutnya.



Gambar 11.7 – desain stop and wait ARQ

Algoritma yang dijalankan pengirim :

```

1  Sn = 0;                                // frame 0 hrs dikirim dulu
2  canSend = true;                        // ijin request pertama dikirim
3  while(true)                            // ulangi selamanya
4  {
5      WaitForEvent();                    // sleep sampai terjadi event
6      if(Event(RequestToSend) AND canSend)
7      {
8          GetData();
9          MakeFrame(Sn);                 // nomor urut Sn
10         StoreFrame(Sn);                // simpan salinannya
11         SendFrame(Sn);                 // kirim frame
12         StartTimer();
13         Sn = Sn + 1;
14         canSend = false;
15     }
16     WaitForEvent()                      // sleep sampai terjadi event
17     if(Event(ArrivalNotification)) // ACK diterima
18     {
19         ReceiveFrame(ackNo); // frame ACK diterima
20         if(Not corrupted AND ackNo == Sn) // ACK valid
21         {
22             StopTimer();
23             PurgeFrame(Sn-1);           // frame tidak perlu lagi
24             canSend = true;
25         }
26     }
27     If(Event(TimeOut))                  // timer expired
28     {
29         StartTimer();
30         ResendFrame(Sn-1);              // kirim ulang salinan
31     }
32 }
33

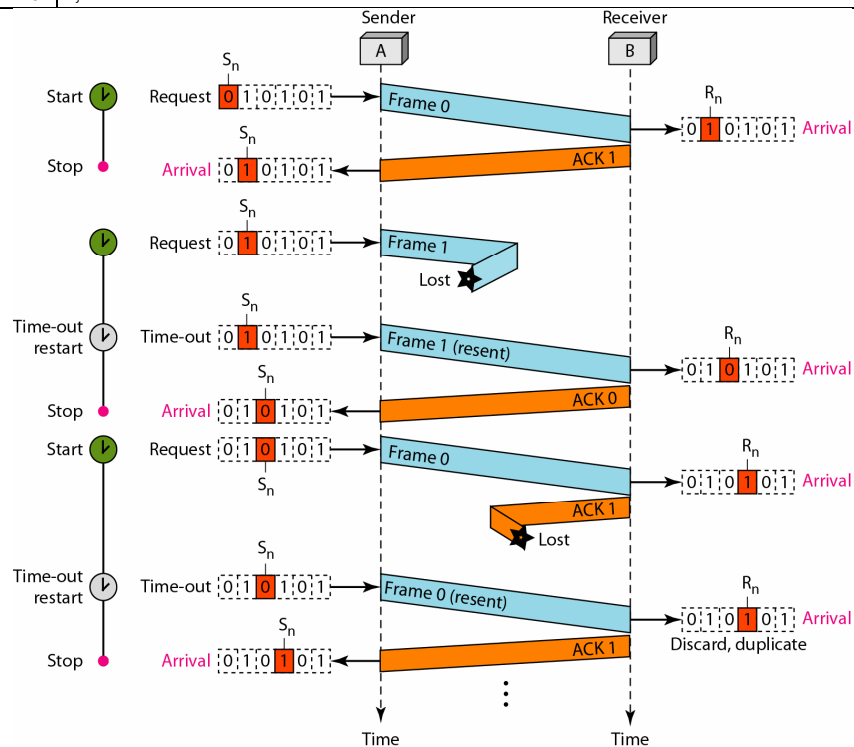
```

Algoritma yang dijalankan penerima :

```

1  Rn = 0; // frame 0 diharapkan dtg dulu
2  while(true) // ulangi selamanya
3  {
4      WaitForEvent(); // sleep sampai terjadi event
5      if(Event(ArrivalNotification)) // Frame data diterima
6      {
7          ReceiveFrame();
8          If(corrupted(frame));
9          Sleep();
10         If(seqNo == Rn) // no frame valid
11         {
12             ExtractData();
13             DeliverData(); // kirim data ke network layer
14             Rn= Rn + 1;
15         }
16         SendFrame(Rn); // kirim frame ACK
17     }
18 }

```



Gambar 11.8 – Contoh aliran data dalam stop and wait ARQ

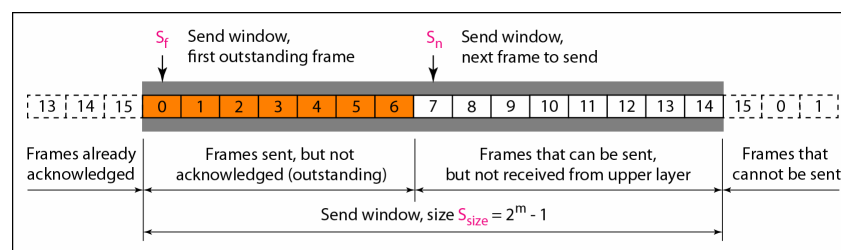
Go back N ARQ

Dalam go back N ARQ, kita dapat mengirim W buah frame, sebelum menerima ACK. Kita tetap mengirimkan frame-frame sampai ACK diterima. Prosedur ini memerlukan fitur tambahan pada stop and wait ARQ.

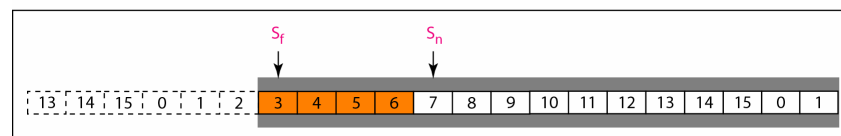
- Nomor urut. Frame pada saat hendak dikirim diberi nomor urut.
- Sliding Window pengirim. Untuk menyimpan frame-frame sebelum di-ACK, kita gunakan konsep window. Kita bayangkan semua frame tersebut disimpan dalam buffer. Frame-frame tersebut disisipkan dalam window.

Frame-frame ke sebelah kiri window adalah yang sudah di-ACK dan dapat dibuang, yang ke kanan jendela tidak dapat dikirim sampai window bergeser ke arahnya. Ukuran window paling besar 2^m-1 . Ukuran window dalam protokol ini tetap (dalam TCP dapat berubah ukuran). Window bergeser ke arah frame-frame yang belum dikirim ketika ACK diterima. Jadi windownya adalah sliding window/ jendela geser.

- Sliding Window penerima. Ukuran window pada penerima dalam protokol ini adalah 1. penerima selalu mencari frame tertentu yang tiba pada urutan tertentu. Frame yang datang tidak terurut diaaikan, dan perlu dikirim ulang lagi. Window penerima juga bergeser seperti window di sisi pengirim.
- Variabel kontrol. Di sisi pengirim ada tiga variabel, yaitu S , S_F dan S_L . S menyimpan nomor urut frame yang sedang dikirim, S_F menyimpan nomor urut frame pertama dalam windows dan S_L menyimpan nomor urut frame terakhir di dalam window. Jika ukuran window adalah W , maka $W=S_L-S_F+1$. Sedangkan di penerima hanya ada variabel R , yang menyimpan nomor urut frame yang hendak diterima. Jika nomor frame yang diterima sama dengan R , maka frame tersebut diterima. Jika nomor frame yang diterima berbeda dengan R maka frame ditolak.
- Timer. Pengirim menset timer untuk setiap frame yang dikirim, sedangkan pada penerima tidak ada timer.
- Acknowledgement. Penerima mengirim ACK positif jika frame telah tiba dengan selamat dan dengan urutan seharusnya. Jika frame rusak waktu diterima atau tidak dalam urutan yang seharusnya, penerima diam dan mengabaikan semua frame yang tidak sesuai urutan sampai dengan ia menerima frame yang sesuai urutan. Jadi pengirim harus mengirimkan ulang frame tersebut karena sampai waktu time out, ACK belum diterima.
- Frame yang dikirim ulang. Ketika frame rusak pengirim mengirim lagi sekumpulan frame mulai frame yang rusak. Itulah sebabnya protokol ini dinamakan go back N ARQ.

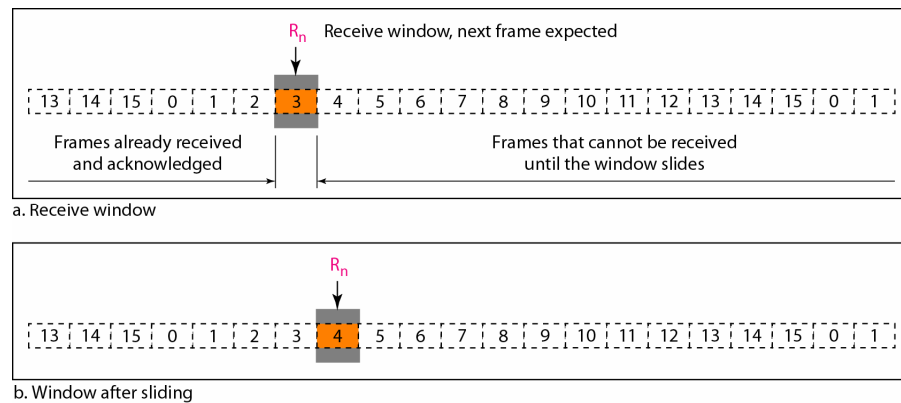


a. Send window before sliding



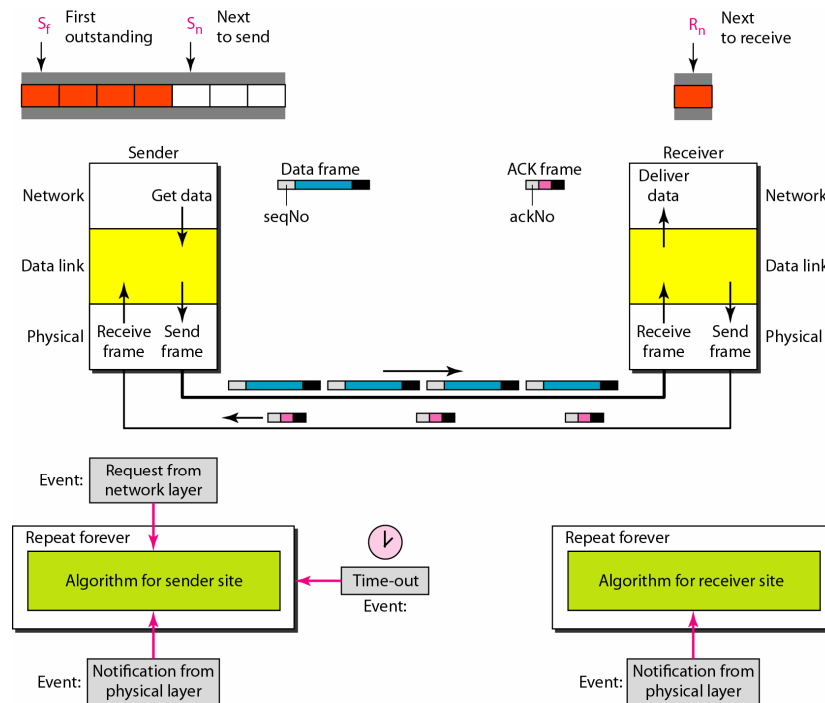
b. Send window after sliding

Gambar 11.9 – sliding window di sisi pengirim



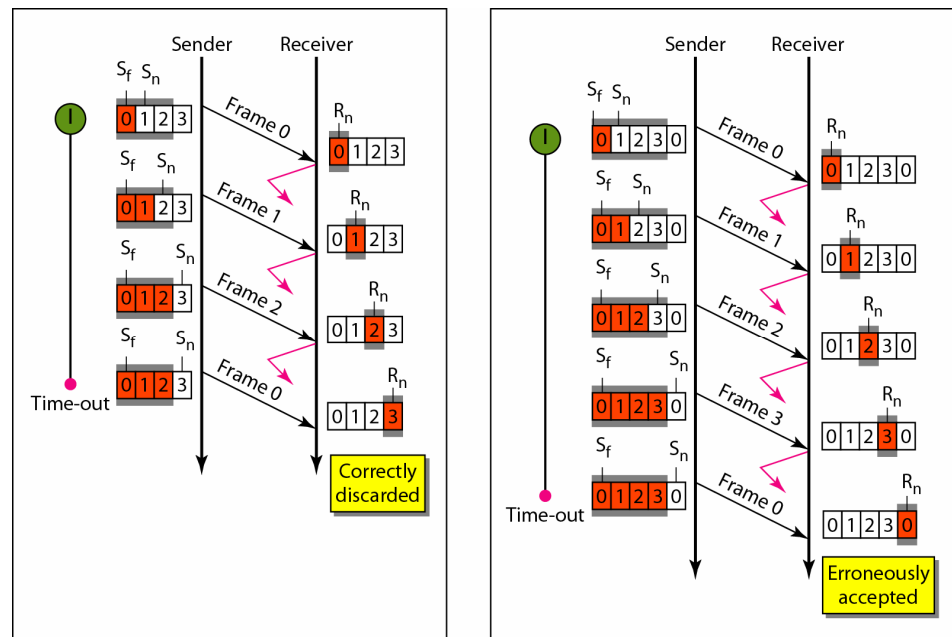
Gambar 11.10 – sliding window sisi penerima

Disain protokol go back N ARQ adalah :



Gambar 11.11 – disain go back N ARQ

Dalam Go back N ARQ, ukuran window pengirim harus lebih kecil dari 2^m , sedangkan ukuran window penerima selalu 1.



a. Window size $< 2^m$

b. Window size $= 2^m$

Gambar 11.12 – ilustrasi go back N ARQ

Dalam operasi normal, pengirim mencatat frame-frame yang dikirim dan mengupdate variabel dan window yang telah di-ACK.

Algoritma pada sisi pengirim adalah :

```

1  Sw =  $2^m - 1$ ;
2  Sf = 0;
3  Sn = 0;
4
5  While(true)                                // ulangi terus
6  {
7      WaitForEvent();
8      If(Event(RequestToSend)) // jika paket hendak dikirim
9      {
10         If(Sn-Sf >= Sw)        // jika window penuh
11             Sleep();
12         GetData();
13         MakeFrame(Sn);
14         StoreFrame(Sn);
15         SendFrame(Sn);
16         Sn = Sn + 1;
17         If(timer not running)
18             StartTimer();
19     }
20     If(event(ArrivalNotification)) // ACK dtg
21     {
22         Receive(ACK);
23         If(corrupted(ACK))
24             Sleep();
25         If((ackNo>Sf)&&(ackNo<=Sn)) // jika ACK valid
26             While(Sf <= ackNo)
27             {
28                 PurgeFrame(Sf);
29                 Sf = Sf + 1;
30             }

```

```

31     StopTimer();
32 }
33
34 If(event(TimeOut))           // timer expires
35 {
36     StartTimer();
37     Temp = Sf;
38     While(Temp < Sn)
39     {
40         SendFrame(Sf);
41         Sf = Sf + 1;
42     }
43 }
44 }

```

Sedangkan algoritma pada sisi penerima adalah :

```

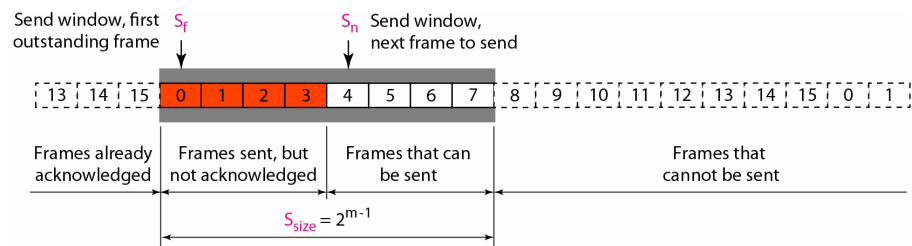
1  Rn = 0;                      // frame 0 diharapkan dtg dulu
2
3  while(true)                  // ulangi selamanya
4  {
5      WaitForEvent();          // sleep sampai terjadi event
6      if(Event(ArrivalNotification)) // Frame data diterima
7      {
8          ReceiveFrame();
9          If(corrupted(frame));
10         Sleep();
11         If(seqNo == Rn)       // no frame valid
12         {
13             DeliverData();    // kirim data ke network layer
14             Rn= Rn + 1;
15             SendACK(Rn);
16         }
17     }
18 }

```

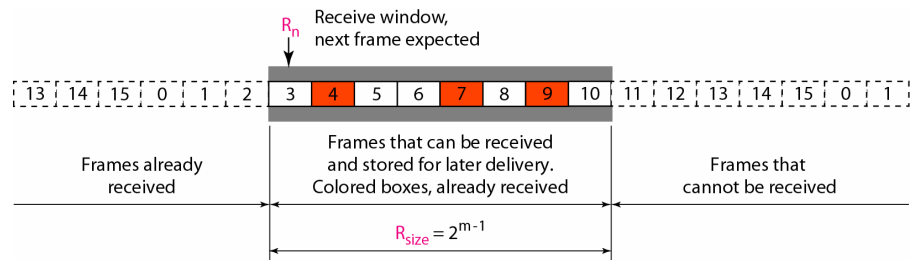
Selective Repeat ARQ

Selective repeat ARQ berbeda dengan go back N ARQ, frame yang dikirim ulang dalam selective repeat ARQ adalah frame yang rusak/ hilang saja. Selective repeat ARQ lebih efisien, tetapi algoritma/ proses dalam penerima lebih kompleks dibandingkan go back N ARQ.

Konfigurasi pengirim dan variabel-variabel kontrolnya sama seperti go back N ARQ. Ukuran windownya pengirim dan penerima harus paling besar setengah kali 2^m . Jadi pada penerima harus ada variabel kontrol seperti di pengirim, yaitu R_F dan R_L untuk mendefinisikan batas kiri dan kanan window.

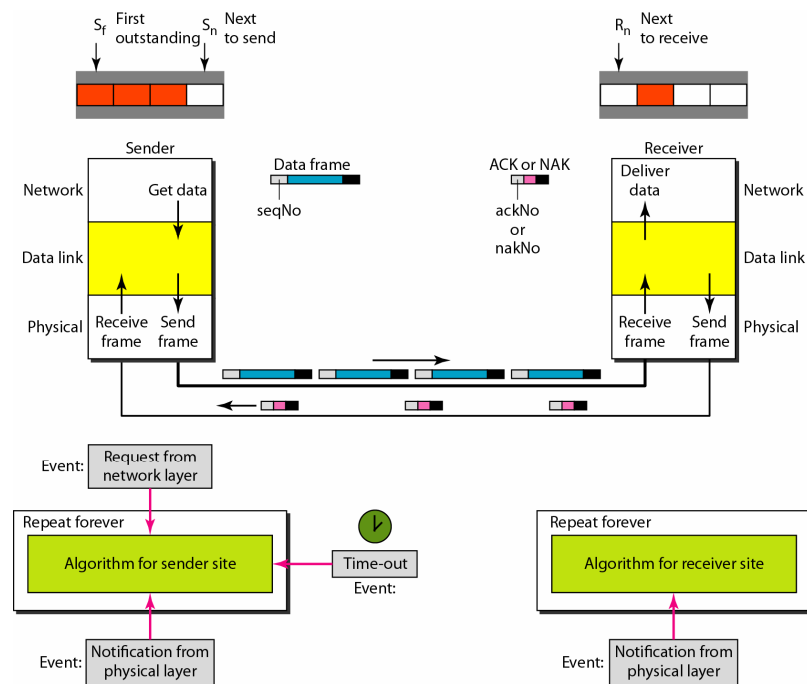


Gambar 11.13 – window pengirim dalam selective repeat ARQ



Gambar 11.14 – window penerima dalam selective repeat ARQ

Dalam selective repeat ARQ juga terdapat negative ACK (NACK) yang melaporkan nomor frame yang rusak.



Gambar 11.15 – desain selective repeat ARQ

Algoritma pada pengirim adalah :

1	$Sw = 2^m - 1;$
2	$Sf = 0;$
3	$Sn = 0;$
4	
5	While(true) // ulangi terus

```

6 {
7   WaitForEvent();
8   If(Event(RequestToSend)) // jika paket hendak dikirim
9   {
10    If(Sn-Sf >= Sw)           // jika window penuh
11    Sleep();
12    GetData();
13    MakeFrame(Sn);
14    StoreFrame(Sn);
15    SendFrame(sn);
16    Sn = Sn + 1;
17    StartTimer(Sn);
18  }
19  If(event(ArrivalNotification)) // ACK dtg
20  {
21    Receive(frame);           // terima ACK atau NAK
22    If(corrupted(frame))
23    Sleep();
24    If(FrameType == NAK)
25    If(nakNo between Sf and Sn)
26    {
27      Resend(nakNo);
28      StartTimer(nakNo);
29    }
30    If(FrameType == ACK)
31    If(nakNo between Sf and Sn)
32    {
33      While(Sf<ackNo)
34      {
35        Purge(Sf);
36        StopTimer(Sf);
37        Sf = Sf +1;
38      }
39    }
40  }
41  If(event(TimeOut(t)))       // timer expires
42  {
43    StartTimer(t);
44    SendFrame(t);
45  }
46 }
47 }

```

Algoritma pada sisi penerima adalah :

```

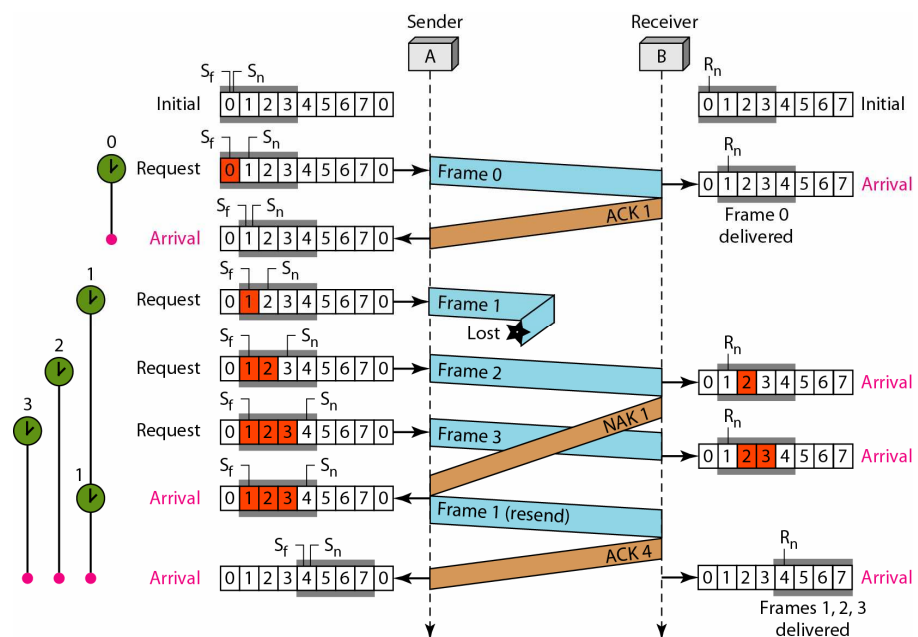
1  Rn = 0;
2  Naksent = false;
3  Ackneeded = false;
4  Repeat(for all slots)
5    Marked(slot) = false;
6
7  while(true)                // ulangi selamanya
8  {
9    WaitForEvent();           // sleep sampai terjadi event
10   if(Event(ArrivalNotification)) // Frame data diterima
11   {
12     ReceiveFrame();
13     If(corrupted(frame) && (NOT naksent))
14     {
15       sendNAK(Rn);
16       NAKsent = true;
17       Sleep();

```

```

18     }
19     If(seqNo <> Rn) && (NOT naksent)
20     {
21         sendNAK(Rn);
22         naksent = true;
23         if((seqNo in window) && (!marked(seqno)))
24         {
25             Storeframe(seqno)
26             Marked(seqno) = true;
27             While(marked(Rn))
28             {
29                 DeliverData(Rn);
30                 Purge(Rn);
31                 Rn = Rn + 1;
32                 Ackneeded = true;
33             }
34             If(ackneeded)
35             {
36                 Sendack(Rn);
37                 Ackneeded = false;
38                 Naksent = false;
39             }
40         }
41     }
42 }
43 }

```



Gambar 16 – Contoh aliran data dalam selective repeat ARQ

Contoh dalam gambar di atas menguraikan kasus frame yang hilang. Perbedaan dengan go back N ARQ adalah timernya, dalam selective repeat semua frame memiliki timer, sehingga timernya perlu diberi nomor. Timer untuk frame 0 dimulai pada request pertama, dan berhenti ketika ACK untuk frame tersebut diterima. Timer untuk frame 1 dimulai ketika request ke dua, dimulai kembali ketika NAK diterima, dan berhenti ketika ACK diterima. Demikian pula untuk frame-frame berikutnya.

Pada penerima, kita perlu membedakan penerimaan frame dan pengirimannya ke layer network. Pada kedatangan ke dua, frame 2 datang, disimpan dan ditandai, tetapi ia tidak dapat dikirim karena frame 1 hilang. Kedatangan berikutnya, frame 3 datang, ditandai dan disimpan, tetapi tidak ada frame yang bisa dikirim. Hanya pada kedatangan terakhir, ketika frame 1 datang, maka frame 1, 2 dan 3 dapat dikirim ke layer network. Ada dua kondisi yang harus dipenuhi untuk mengirim frame ke layer network, pertama kumpulan frame harus telah tiba, kedua kumpulan frame dimulai dari awal window.

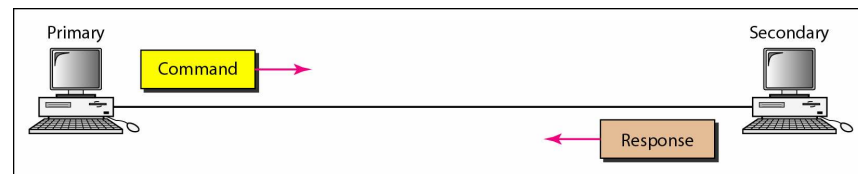
High Level Data Link Control (HDLC)

HDLC adalah protokol bit oriented untuk komunikasi point to point atau multipoint. Diimplementasikan pada mekanisme ARQ.

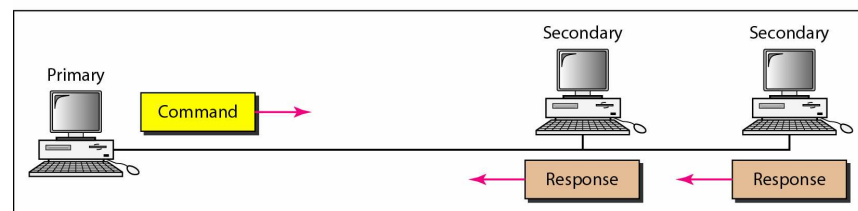
Konfigurasi dan Mode Transfer

Ada dua mode yang digunakan, yaitu NRM/ normal response mode dan ABM/ asynchronous balanced mode.

NRM. Dalam NRM konfigurasi station tidak seimbang. Kita memiliki satu station primer dan beberapa station sekunder. Station primer dapat mengirim perintah, sedangkan station sekunder hanya mengirim respon. NRM digunakan pada link point to point dan multipoint.



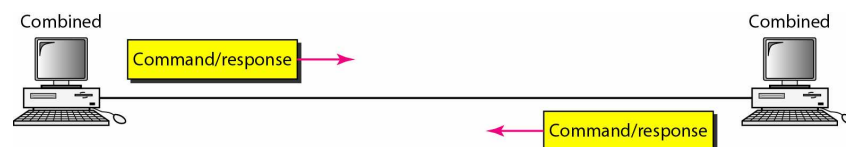
a. Point-to-point



b. Multipoint

Gambar 11.17 - NRM

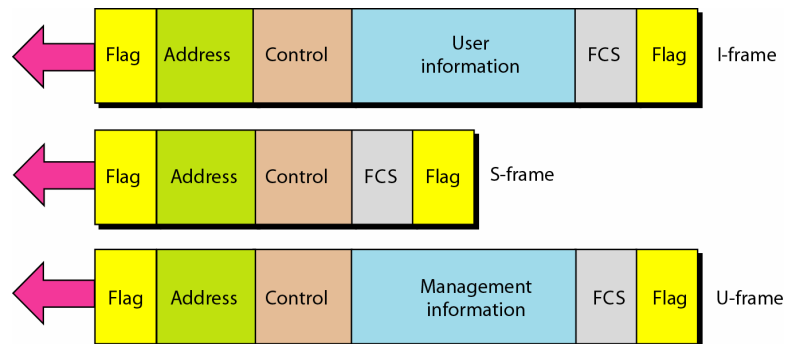
ABM. Dalam ABM konfigurasi station seimbang. Link yang digunakan point to point, dan setiap station dapat berfungsi sebagai station primer atau sekunder.



Gambar 18 - ABM

Frame

HDLC mendefinisikan tiga tipe frame, information frame (I-frame), supervisory frame (S-frame), dan unnumbered frame (U-frame). Setiap tipe frame bekerja sebagai amplop bagi transmisi bagi tipe pesan yang berbeda. I-frame digunakan untuk mengirim data user dan informasi kontrol yang berhubungan dengan data user (piggybacking). S-frame digunakan hanya untuk mengirim informasi kontrol. U-frame dicadangkan untuk manajemen sistem. Informasi yang ada dalam U-frame digunakan untuk mengelola link itu sendiri.



Gambar 11.19 – frame HDLC

Setiap frame dalam HDLC dapat berisi sampai dengan 6 field (kecuali S-frame memiliki 4 field).

Field flag berisi 8 bit berurutan dengan pola 01111110 yang mengidentifikasi baik awal dan akhir frame dan melayani sebagai pola sinkronisasi bagi penerima.

Field address berisi alamat station sekunder yang berisi asal atau tujuan frame. Jika station primer membuat frame, ia berisi alamat tujuan. Jika station sekunder membuat frame, ia berisi alamat sumber. Panjangnya dapat 1 byte atau beberapa byte, bergantung kebutuhan jaringan. Satu byte dapat mengidentifikasi sampai dengan 128 station (1 bit digunakan untuk keperluan lain). Jika panjangnya 1 byte, bit terakhirnya selalu 1, semua byte selain terakhir diakhiri 0, hanya byte terakhir diakhiri 1. diakhiri 0 artinya masih ada byte address. Jaringan yang tidak menggunakan konfigurasi primer/ sekunder, seperti Ethernet memiliki dua field address/ alamat, yaitu alamat pengirim dan penerima.

Field kontrol adalah segmen 1 atau 2 byte, digunakan untuk flow control dan error control. Interpretasi bit-bit dalam field ini berbeda-beda untuk tipe frame yang berbeda.

Field informasi berisi data user dari layer network atau informasi manajemen jaringan. Panjangnya dapat bervariasi dari satu jaringan ke jaringan lain, tetapi selalu tetap untuk tiap jaringan.

Field FCS/ frame check sequence digunakan untuk deteksi error.

I-frame

I-Frame didisain untuk membawa data user dari network layer. Di dalamnya terdapat informasi flow control dan error control (piggybacking).

Bit pertama dalam I-frame adalah bit 0, artinya ia adalah I-frame.

3 bit berikutnya disebut N(S), mendefinisikan nomor urut frame dalam perjalanan. Dengan panjang 3 bit, maka nomor urutnya adalah 0 s.d. 7.

Bit berikutnya dinamakan P/F, yang memiliki arti jika ia bernilai 1, dan berarti poll atau final. Berarti poll jika frame dikirim oleh primary station ke secondary station. Berarti final jika frame dikirim dari secondary ke primary station.

3 bit terakhir adalah N(R), berubungan dengan nilai ACK ketika menggunakan piggybacking.

S-frame

Supervisory frame digunakan untuk flow control dan error control ketika piggybacking tidak memungkinkan. S-frame tidak memiliki field informasi.

2 bit pertama adalah 10, menandakan S-frame.

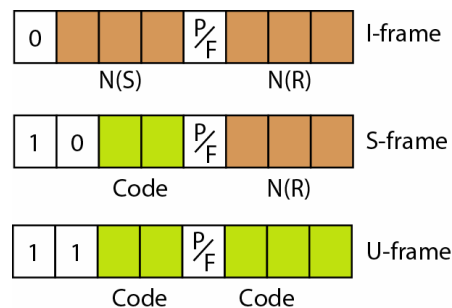
2 bit berikutnya adalah control field, berupa kode yang mendefinisikan 4 tipe S-frame, yaitu receive ready (RR) berisi 00, receive not ready (RNR) berisi 10, reject (REJ) berisi 01, dan selective reject (SREJ) berisi 11.

Bit ke 5, berupa P/F, gunanya seperti pada I-frame.

3 bit terakhir adalah N[®], berhubungan dengan nilai ACK dan NAK.

U-frame

U-frame digunakan untuk manajemen pertukaran sesi dan informasi kontrol antar perangkat.



Gambar 11.20 Control field untuk I-frame, S-frame dan U-frame

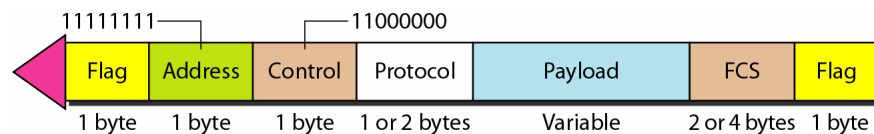
U-frame berisi information field yang diisi informasi manajemen sistem. Kode dalam U-frame terdiri dari 2 bagian, masing-masing 2 bit, sebelum P/F (prefix) dan sesudah P/F (suffix).

<i>Kode</i>	<i>Command</i>	<i>Response</i>	<i>Arti</i>
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, diperluas
11 100	SABM	DM	Set asynchronous balanced mode atau disconnect mode
11 110	SABME		Set asynchronous balanced mode, diperluas
00 000	UI	UI	Informasi tak bernomor
00 110		UA	Acknowledgement tak bernomor
00 010	DISC	RD	Disconnect atau request disconnect
10 000	SIM	RIM	Set initialization mode atau request information mode
00 100	UP		Poll tak bernomor
11 001	RSET		Reset
11 101	XID	XID	Exchange ID
10 001	FRMR	FRMR	Frame reject

Point to Point Protocol (PPP)

Meskipun HDLC adalah protokol umum yang dapat digunakan dalam konfigurasi point to point maupun multipoint, protokol yang paling umum adalah Point to Point Protocol (PPP) yang adalah protokol byte oriented.

Saat ini sebagian besar pengguna Internet melakukan koneksi melalui PPP, menggunakan modem tradisional, modem DSL, atau modem kabel.



Gambar 11.21 – format frame dalam PPP

Format frame dalam PPP terdiri dari :

Field flag, seperti HDLC mengidentifikasi batas frame PPP, isinya 01111110.

Field address. Karena PPP digunakan untuk koneksi point to point, ia menggunakan alamat broadcast HDLC yaitu 11111111 untuk mengabaikan alamat data link dalam protokol.

Field control, menggunakan format U-frame dari HDLC. Nilainya 11000000 untuk menunjukkan bahwa frame tidak berisi nomor urut dan tidak ada flow control dan error control.

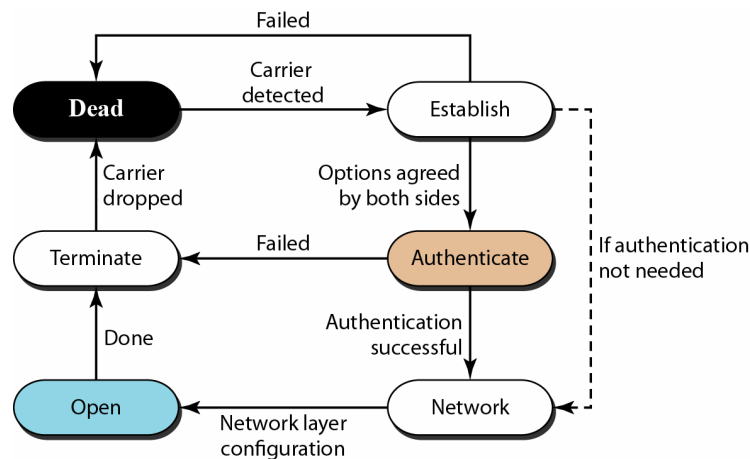
Field protocol, mendefinisikan apa yang sedang dibawa dalam field data apakah data user atau informasi lain.

Field data berisi data user atau informasi lain.

Field frame check sequence/ FCS. Seperti dalam HDLC berupa CRC 2 byte atau 4 byte.

Fasa Transisi

Koneksi PPP melalui berbagai fasa berbeda yang digambarkan dalam diagram perubahan keadaan/ transition state berikut.



Gambar 11.22 – diagram transition state

Dead state, artinya link tidak sedang digunakan. Tidak ada carrier yang aktif dan line kosong.

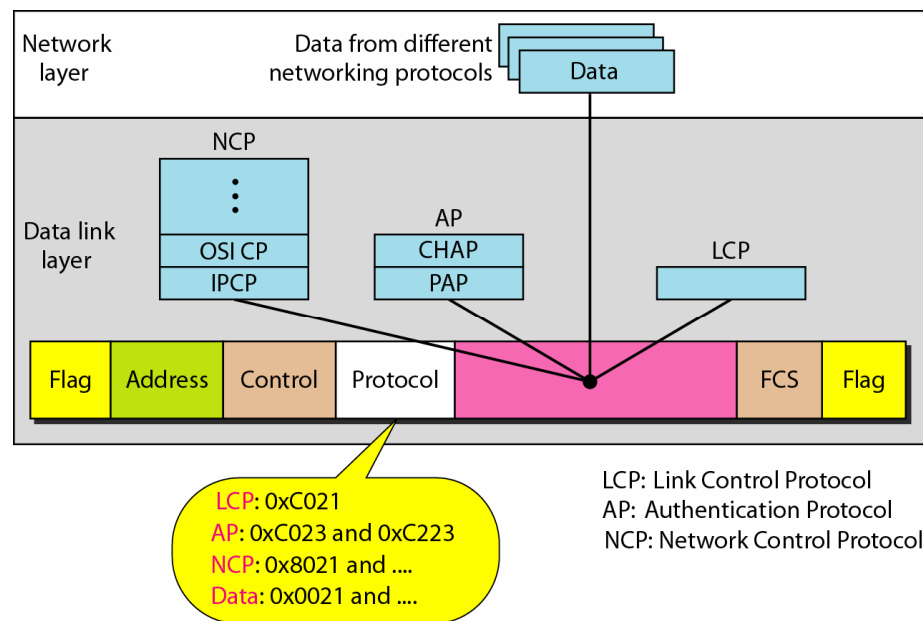
Establish state. Ketika salah satu titik memulai komunikasi, koneksi berubah ke establish state. Dalam state ini, terjadi negosiasi antar dua pihak/ party. Jika negosiasi berhasil, sistem berpindah ke authenticate state (jika perlu otentikasi) atau langsung ke network state. Berberapa paket bertukar dalam keadaan ini.

Authenticate state. State ini opsional. Kedua belah pihak yang menentukan apakah melalui state ini atau tidak. Jika perlu otentikasi, terjadi paket-paket otentikasi. Jika otentikasi berhasil maka sistem berpindah ke network state, jika tidak ia berpindah ke terminate state.

Network state. State ini adalah inti dari perubahan keadaan. Setelah koneksi berhasil maka pertukaran paket data dan kontrol dimulai. Pertukaran ini dilakukan dalam open state.

Open state. Ketika koneksi telah mencapai keadaan ini, pertukaran kontrol dan data dapat dimulai. Koneksi tetap ada dalam keadaan ini sampai salah satu pihak mengakhiri koneksi.

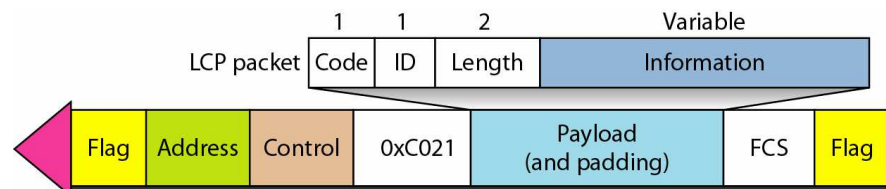
Terminate state. Ketika koneksi mencapai tahap ini, pertukaran paket data dan kontrol antara dua pihak dan telah berakhir dan link ditutup.



Gambar 11.23 – PPP Stack

Meskipun PPP adalah protocol data link, PPP menggunakan protokol-protokol lain untuk membuka link, untuk melakukan otentikasi dan untuk membawa data ke layer network.

Link Control Protocol (LCP) bertanggung jawab untuk membuka, memelihara, mengkonfigurasi dan mengakhiri link. Ia juga yang menyediakan mekanisme negosiasi untuk menset opsi antara dua pihak. Jadi jika PPP berisi paket LCP artinya ia mungkin ada dalam establish state atau terminate state.

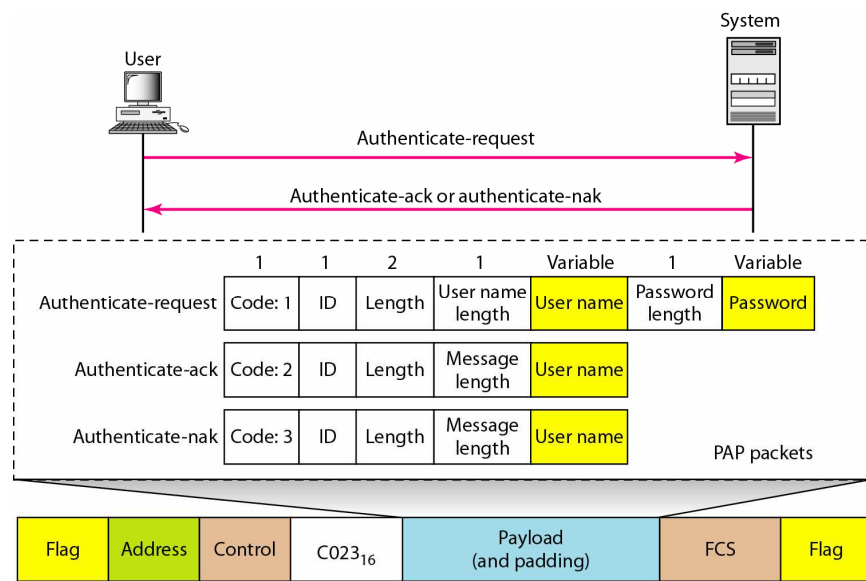


Gambar 11.24 – paket LCP

Protokol otentikasi. Otentikasi berarti melakukan validasi identitas user yang perlu mengakses sumber daya. PPP telah membuat dua protokol otentikasi yaitu password authentication protocol (PAP) dan challenge handshake authentication protocol (CHAP).

PAP adalah protokol sederhana dengan dua tahap proses, yaitu :

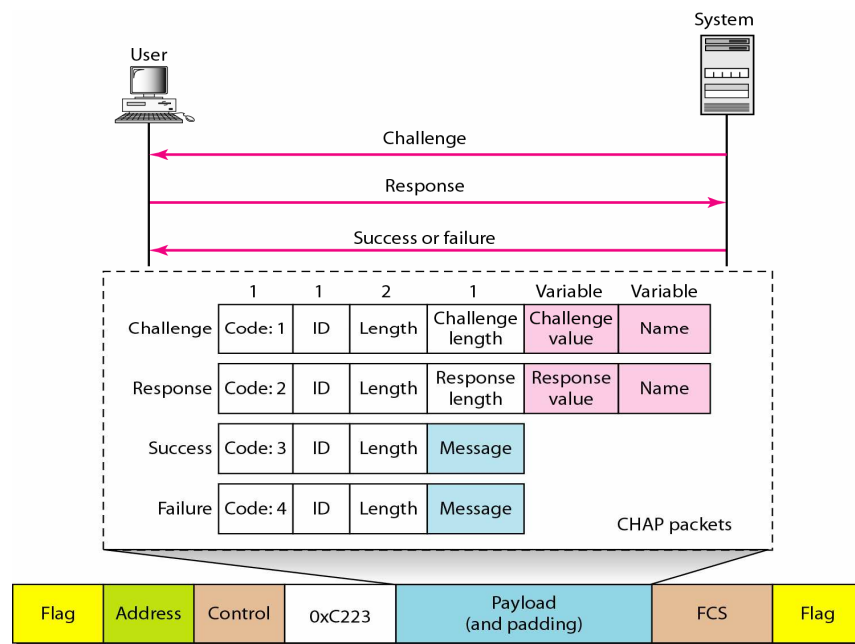
- User yang ingin akses ke sistem mengirim identifikasi otentikasi, biasanya berupa nama user dan password
- Sistem memeriksa validitas identitas dan password, lalu menerima atau menolak koneksi



Gambar 11.25 – paket PAP

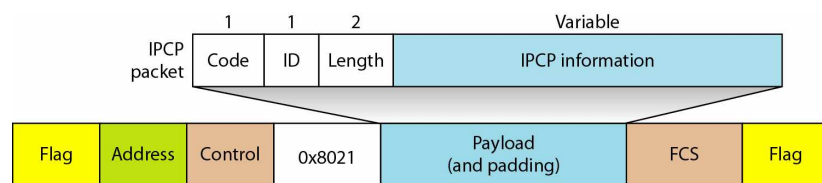
CHAP adalah protokol otentikasi three-way handshake, yang lebih aman daripada PAP. Tahapannya adalah :

- Sistem mengirim ke user paket challenge berisi nilai challenge biasanya beberapa byte.
- User mengaplikasikan fungsi yang mengambil nilai challenge, password dan memperoleh hasil. User mengirim hasil ke sistem.
- Sistem melakukan hal yang sama. Dengan fungsi yang sama terhadap password user tersebut dan nilai challenge untuk memperoleh hasil. Hasil ini dibandingkan dengan hasil yang diperoleh dari user. Jika sama akses diterima, jika tidak maka akses ditolak.



Gambar 11.26 – paket CHAP

NCP dipanggil setelah link terbuka dan otentikasi (jika ada) berhasil. NCP adalah kumpulan protokol kontrol untuk memungkinkan enkapsulasi data dari layer network ke dalam frame PPP. Salah satu protokolnya adalah IPCP/ internetwork protocol control protocol. Protokol ini digunakan untuk membuka dan menutup koneksi pada layer network.



Gambar 11.27 – paket IPCP