

La capa de enlace y las redes de área local

Juan Manuel Orduña Huertas

Telemática y Sistemas de Transmisión de Datos - Curso
2011/2012

Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Medio multidifusión

Primeras LAN (IEEE 802.3):

- 1 Sólo una máquina puede estar transmitiendo a la vez.
- 2 De lo contrario, colisión
- 3 No hay canales secundarios
- 4 Control de acceso distribuido

Tipos

- De contienda
- Reserva
- Rotación circular

Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

CSMA 1-persistente

- ❶ La estación escucha el medio (comprueba portadora).
 - ❷ Si está libre (no hay portadora) entonces transmite una trama.
 - ❸ Si canal ocupado, espera hasta que portadora desaparece. Entonces, transmite una trama.
 - ❹ Si colisión, espera un tiempo aleatorio y comienza de nuevo el algoritmo.
- Principal parámetro de rendimiento: retardo de propagación.
 - Existen colisiones aunque el retardo sea nulo

CSMA No persistente

- ➊ La estación escucha el medio (comprueba portadora).
 - ➋ Si está libre (no hay portadora) entonces transmite una trama.
 - ➌ Si canal ocupado, espera tiempo aleatorio y comienza de nuevo.
- Mejor utilización del canal, pero mayor retardo en el envío de tramas.

CSMA p-persistente

- 1 Tiempo ranurado.
- 2 La estación escucha el medio (comprueba portadora).
- 3 Si está libre (no hay portadora) transmite una trama con probabilidad p y se espera hasta siguiente trama con probabilidad $q = 1 - p$.
- 4 Si en esa ranura el canal también está libre, la estación transmite o se espera nuevamente con probabilidades p y q , respectivamente.
- 5 El proceso se repite hasta que la trama se transmite o hasta que el canal está ocupado. En este último caso, la estación espera un tiempo aleatorio y comienza de nuevo el algoritmo.

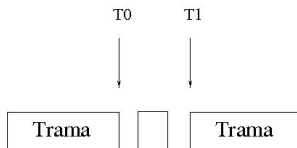
Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Diferencias con CSMA

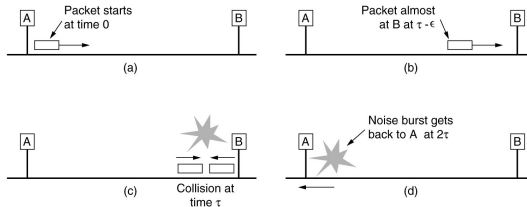


a) Colision CSMA



b) Colision CSMA/CD

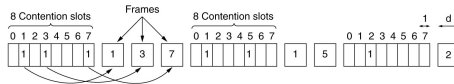
Detección de colisión en el peor caso



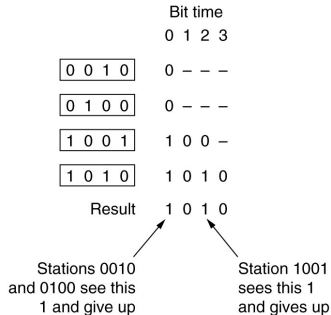
Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

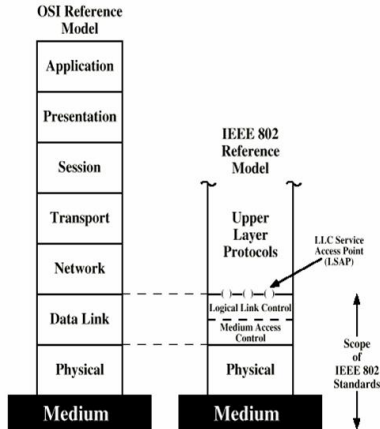
Protocolo de mapa de bits



Protocolo de cuenta atrás binaria



Modelo de referencia IEEE 802



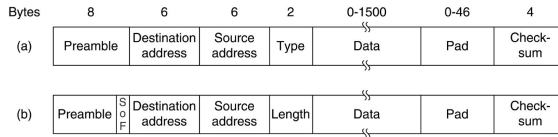
Algunos comités de trabajo IEEE 802

- 802: Arquitectura y cuestiones generales
- 802.1: Aspectos comunes: puentes, redes locales virtuales, etc.
- 802.2: Logical Link Control (LLC)
- 802.3: Redes CSMA/CD (Ethernet)
- 802.5: Redes Token Ring
- 802.7: Grupo asesor en redes de banda ancha
- 802.8 Grupo asesor en fibras ópticas
- 802.9: Redes de servicios Integrados (Iso-Ethernet)
- 802.10: Seguridad en LAN/MAN
- 802.11: LANs inalámbricas
- 802.12: Redes Demand Priority (100VG-AnyLAN)
- 802.14: Redes de TV por cable
- 802.15: Redes Bluetooth

Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Subcapa MAC de la 802.3: Formatos de trama



Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Diferencias

- Extensión de la portadora: La trama resultante tiene una duración equivalente a 4.096 bits
- Ráfagas de tramas: Permitir la transmisión consecutiva de varias tramas cortas sin necesidad de dejar el control del CSMA/CD.

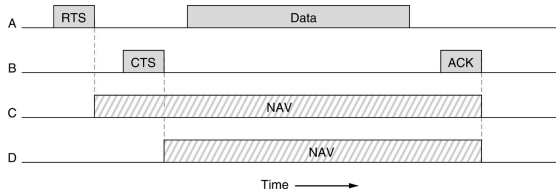
Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

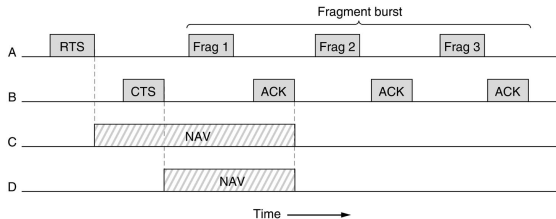
Problemas de la estación expuesta y estación oculta

A $\longleftrightarrow_{10m.}$ B $\longleftrightarrow_{10m.}$ C $\longleftrightarrow_{10m.}$ D

Detección de canal virtual usando CSMA/CA



Ráfagas de fragmentos



Problemática de la capa de enlace

Función: Lograr una comunicación fiable y eficiente entre dos máquinas adyacentes, donde adyacente quiere decir que están conectadas por un canal que entrega los bits en el mismo orden que los recibe.

Problemas:

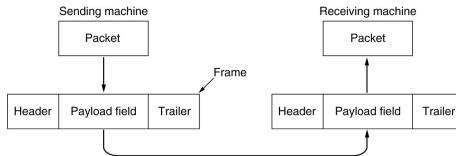
- El circuito de comunicación puede generar errores.
- La velocidad de transmisión es finita.
- Hay un retardo de propagación inherente al canal.
- Un emisor rápido no debe saturar a uno lento.

Contenido

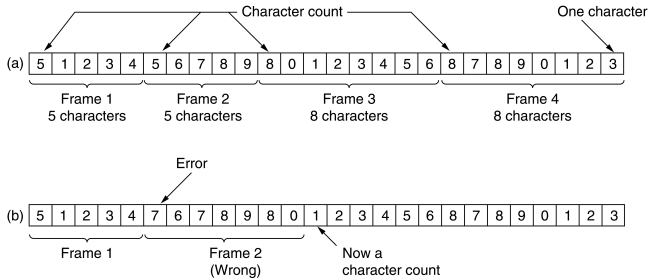
- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 **La Subcapa de enlace**
 - **Parámetros de diseño de la capa de enlace**
 - **Detección de errores**
 - **Protocolos elementales de la capa de enlace**
 - **Protocolos de ventana deslizante**
 - **Ejemplos de protocolos de enlace de datos**

Funciones

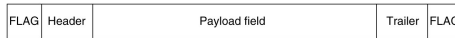
- Proporcionar un interfaz bien definido con la capa de red.
- Detectar (y opcionalmente corregir) los posibles errores del canal de comunicación.
- Controlar el flujo de datos.



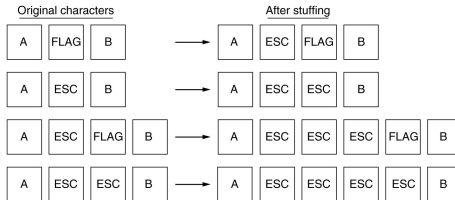
Entramado: Cuenta de caracteres



Entramado: Banderas



(a)



(b)

Entramado: Rellenado de bits

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Detección de errores y control de flujo

- La capa de enlace detecta los errores, pero no reenvía las tramas (excepto X.25).
- Puede hasta desaparecer una trama completa → Temporizadores
- Necesaria realimentación con tramas de confirmación.
- Puede desaparecer confirmación → Números de secuencia

Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 **La Subcapa de enlace**
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Códigos Cíclicos Redundantes

- Cadenas de bits = representaciones de polinomios con coeficientes 0 y 1.
- Trama de k bits = polinomio de k términos, (de izquierda a derecha) = x^{k-1} a x^0
- Ejemplo: cadena 111010 representa al polinomio $x^5 + x^4 + x^3 + x^1$
- Aritmética polinomial o módulo 2: suma y resta de polinomios mediante una OR-Exclusiva.
- División de la misma forma que en binario, excepto que la resta es (OR-Exclusiva). Un divisor cabe en un dividendo si este tiene tantos bits significativos como el divisor.

Polinomio generador

- Tanto el emisor como el receptor deben ponerse de acuerdo en el **polinomio generador**, $G(x)$.
- Los bits de mayor y de menor orden de este polinomio deben ser un 1.
- Para calcular el CRC de una trama de m bits, correspondiente al polinomio $M(x)$, la trama debe ser más larga que el polinomio generador.
- Idea: una suma de verificación (checksum o CRC) al final de la trama de tal manera que el polinomio representado por la trama con suma de verificación sea divisible entre $G(x)$.
- Cuando el receptor recibe la trama con suma de verificación, intenta dividirla entre $G(x)$. Si hay un resto, ha habido un error de transmisión.

Algoritmo para calcular CRC

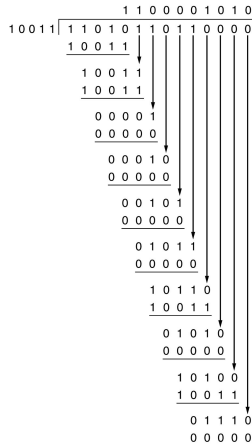
- 1 - Sea r el grado de $G(x)$. Hay que añadir r bits cero al final de la trama, para que ahora contenga $m + r$ bits y corresponda al polinomio $M(x)x^r$.
- 2 - Dividir la cadena de bits correspondiente a $M(x)x^r$ entre la correspondiente a $G(x)$.
- 3 - Restar el resto de la división anterior (que siempre es de r o menos bits) a la cadena de bits correspondiente a $M(x)x^r$ usando una resta módulo 2. El resultado es la trama con CRC que debe transmitirse, $T(x)$.

Cálculo para trama 1101011011 y $G(x) = x^4 + x + 1$

Frame : 1101011011

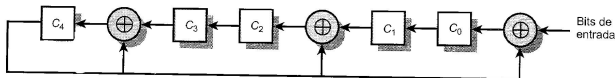
Generator: 10011

Message after 4 zero bits are appended: 11010110110000



Remainder

Circuito detector de errores y Cálculo de un CRC



= Registro de desplazamiento de 1 bit

$\oplus$ = Circuito OR-exclusivo

(a) Implementación mediante registro de desplazamiento

	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3$	$C_4 \oplus C_1$	$C_4 \oplus \text{entrada}$	entrada
Inicialización	0	0	0	0	0	0	0	1	1
Paso1	0	0	0	0	1	0	0	0	0
Paso2	0	0	0	1	0	0	1	1	1
Paso3	0	0	1	0	1	0	0	0	0
Paso4	0	1	0	1	0	1	1	0	0
Paso5	1	0	1	0	0	1	1	1	0
Paso6	1	1	1	0	1	0	1	0	1
Paso7	0	1	1	1	0	1	1	1	1
Paso8	1	1	1	0	1	0	1	1	0
Paso9	0	1	1	1	1	1	1	1	1
Paso10	1	1	1	1	1	0	0	1	0
Paso11	0	1	0	1	1	1	1	0	0
Paso12	1	0	1	1	0	1	0	1	0
Paso13	1	1	0	0	1	0	1	1	0
Paso14	0	0	1	1	1	0	1	0	0
Paso15	0	1	1	1	0	1	1	0	—

Mensaje a enviar

Cinco ceros añadidos

(b) Ejemplo con entrada de 1010001101

Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 **La Subcapa de enlace**
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Supuestos

- ❶ - Máquina A desea enviar un flujo de datos a B. A tiene un suministro infinito de datos (su capa de red).
- ❷ - Las máquinas no fallan. Sólo el canal puede fallar
- ❸ - Existen procedimientos como To_physical_layer para enviar una trama y From_physical_layer para recibir una trama. El hardware emisor calcula y agrega el CRC.
- ❹ - Cuando llega una trama al receptor, el hardware calcula el CRC. Si incorrecta, event = checksum_err. Si correcta event = frame_arrival, para que pueda adquirirla mediante from_physical_layer


```
#define MAX_PKT 1024                /* determines packet size in bytes */

typedef enum {false, true} boolean; /* boolean type */
typedef unsigned int seq_nr;        /* sequence or ack numbers */
typedef struct {unsigned char data[MAX_PKT];} packet; /* packet definition */
typedef enum {data, ack, nak} frame_kind; /* frame_kind definition */

typedef struct {                    /* frames are transported in this layer */
    frame_kind kind;                /* what kind of a frame is it? */
    seq_nr seq;                    /* sequence number */
    seq_nr ack;                    /* acknowledgement number */
    packet info;                   /* the network layer packet */
} frame;

/* Wait for an event to happen; return its type in event. */
void wait_for_event(event_type *event);

/* Fetch a packet from the network layer for transmission on the channel. */
void from_network_layer(packet *p);

/* Deliver information from an inbound frame to the network layer. */
void to_network_layer(packet *p);

/* Go get an inbound frame from the physical layer and copy it to r. */
void from_physical_layer(frame *r);

/* Pass the frame to the physical layer for transmission. */
void to_physical_layer(frame *s);

/* Start the clock running and enable the timeout event. */
void start_timer(seq_nr k);

/* Stop the clock and disable the timeout event. */
void stop_timer(seq_nr k);

/* Start an auxiliary timer and enable the ack_timeout event. */
void start_ack_timer(void);

/* Stop the auxiliary timer and disable the ack_timeout event. */
```

Protocolo símplex no restringido

- Los datos se transmiten sólo en una dirección.
- Las capas de red tanto del emisor como del receptor siempre están listas.
- El tiempo de procesamiento puede ignorarse.
- Hay un espacio infinito de búfer.
- Canal de comunicación entre las capas de enlace de datos ni daña ni pierde las tramas.

/* Protocol 1 (utopia) provides for data transmission in one direction only, from sender to receiver. The communication channel is assumed to be error free and the receiver is assumed to be able to process all the input infinitely quickly. Consequently, the sender just sits in a loop pumping data out onto the line as fast as it can. */

```
typedef enum {frame_arrival} event_type;  
#include "protocol.h"
```

```
void sender1(void)  
{  
    frame s;                /* buffer for an outbound frame */  
    packet buffer;           /* buffer for an outbound packet */  
  
    while (true) {  
        from_network_layer(&buffer); /* go get something to send */  
        s.info = buffer;             /* copy it into s for transmission */  
        to_physical_layer(&s);       /* send it on its way */  
    }                                /* Tomorrow, and tomorrow, and tomorrow,  
                                     Creeps in this petty pace from day to day  
                                     To the last syllable of recorded time.  
                                     - Macbeth, V, v */  
}
```

```
void receiver1(void)  
{  
    frame r;  
    event_type event;        /* filled in by wait, but not used here */  
  
    while (true) {  
        wait_for_event(&event);    /* only possibility is frame_arrival */  
        from_physical_layer(&r);    /* go get the inbound frame */  
        to_network_layer(&r.info);  /* pass the data to the network layer */  
    }  
}
```

Protocolo símplex e parada y espera (stop & wait)

- Los datos se transmiten sólo en una dirección.
- Las capas de red tanto del emisor como de] receptor siempre están listas.
- El tiempo de procesamiento puede ignorarse.
- **NO** hay un espacio infinito de búfer.
- Canal de comunicación entre las capas de enlace de datos ni daña ni pierde las tramas.

Evitar que emisor sature a receptor

Protocolo símplex e parada y espera (stop & wait)

Solución = retroalimentación

- Receptor devuelve al emisor trama ficticia que autoriza al emisor para transmitir la siguiente trama.
- Tras haber enviado una trama, el emisor espera que llegue la (es decir, la confirmación de recepción).

Tráfico de datos símplex pero canal semidúplex.

/* Protocol 2 (stop-and-wait) also provides for a one-directional flow of data from sender to receiver. The communication channel is once again assumed to be error free, as in protocol 1. However, this time, the receiver has only a finite buffer capacity and a finite processing speed, so the protocol must explicitly prevent the sender from flooding the receiver with data faster than it can be handled. */

```
typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender2(void)
{
    frame s; /* buffer for an outbound frame */
    packet buffer; /* buffer for an outbound packet */
    event_type event; /* frame_arrival is the only possibility */

    while (true) {
        from_network_layer(&buffer); /* go get something to send */
        s.info = buffer; /* copy it into s for transmission */
        to_physical_layer(&s); /* bye-bye little frame */
        wait_for_event(&event); /* do not proceed until given the go ahead */
    }
}

void receiver2(void)
{
    frame r, s; /* buffers for frames */
    event_type event; /* frame_arrival is the only possibility */
    while (true) {
        wait_for_event(&event); /* only possibility is frame_arrival */
        from_physical_layer(&r); /* go get the inbound frame */
        to_network_layer(&r.info); /* pass the data to the network layer */
        to_physical_layer(&s); /* send a dummy frame to awaken sender */
    }
}
```

Protocolo símplex para un canal con ruido

- Los datos se transmiten sólo en una dirección.
- Las capas de red tanto del emisor como de] receptor siempre están listas.
- El tiempo de procesamiento puede ignorarse.
- **NO** hay un espacio infinito de búfer.
- Canal de comunicación CON ERRORES: daña y/o pierde tramas.

Protocolo símples para un canal con ruido

¿Solución = Temporizador en emisor y confirmación de recepción en receptor?

- ❶ A envía trama 1 a B. B devuelve a A una trama de confirmación de recepción.
- ❷ La trama de confirmación de recepción se pierde por completo. Nunca llega.
- ❸ El temporizador de la capa de enlace de datos de A expira y envía otra vez la trama 1.
- ❹ La trama duplicada llega bien a B. El protocolo falla.

Receptor debe diferenciar entre trama nueva y retransmisión de trama → Números de secuencia


```

/* Protocol 3 (par) allows unidirectional data flow over an unreliable channel. */
#define MAX_SEQ 1          /* must be 1 for protocol 3 */
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"

void sender3(void)
{
    seq_nr next_frame_to_send; /* seq number of next outgoing frame */
    frame s; /* scratch variable */
    packet buffer; /* buffer for an outbound packet */
    event_type event;

    next_frame_to_send = 0; /* initialize outbound sequence numbers */
    from_network_layer(&buffer); /* fetch first packet */
    while (true) {
        s.info = buffer; /* construct a frame for transmission */
        s.seq = next_frame_to_send; /* insert sequence number in frame */
        to_physical_layer(&s); /* send it on its way */
        start_timer(s.seq); /* if answer takes too long, time out */
        wait_for_event(&event); /* frame_arrival, cksum_err, timeout */
        if (event == frame_arrival) {
            from_physical_layer(&s); /* get the acknowledgement */
            if (s.ack == next_frame_to_send) {
                stop_timer(s.ack); /* turn the timer off */
                from_network_layer(&buffer); /* get the next one to send */
                inc(next_frame_to_send); /* invert next_frame_to_send */
            }
        }
    }
}

void receiver3(void)
{
    seq_nr frame_expected;
    frame r, s;
    event_type event;

    frame_expected = 0;
    while (true) {
        wait_for_event(&event); /* possibilities: frame_arrival, cksum_err */
        if (event == frame_arrival) { /* a valid frame has arrived. */
            from_physical_layer(&r); /* go get the newly arrived frame */
            if (r.seq == frame_expected) { /* this is what we have been waiting for. */
                to_network_layer(&r.info); /* pass the data to the network layer */
                inc(frame_expected); /* next time expect the other sequence nr */
            }
            s.ack = 1 - frame_expected; /* tell which frame is being acked */
            to_physical_layer(&s); /* send acknowledgement */
        }
    }
}

```

Contenido

- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 **La Subcapa de enlace**
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

Mejora de prestaciones

Full dúplex = 2 canales separados con tráfico de datos simplex en cada uno

- Se pierde la mitad del ancho de banda.

Solución:

- Un sólo canal y tráfico de datos en ambas direcciones.
- Tramas de datos de A a B se mezclan con tramas de confirmación de recepción de A a B.
- Campo de tipo (kind) en el encabezado de una trama de enviada.
- Otra mejora: piggybacking. Mejora el ancho de banda pero necesita temporizadores.

Ventanas deslizantes

- Ventana de emisor y receptor no tienen mismos límites.
- Debe entregarse a capa de red del destino los paquetes en orden.
- Números secuencia en v. emisión = tramas enviadas sin confirmar.
- Nuevo paquete = número secuencial mayor, si tamaño $\leq$ permitido. Al llegar confirmación, el extremo inferior avanza en 1. Emisor necesita n buffers
- Números secuencia en v. recepción = tramas que se pueden aceptar. Toda otra trama se descarta.
- Nueva trama cuyo N^o sec. = límite inferior $\rightarrow$ avanza la ventana. V. emisión **siempre** mismo tamaño

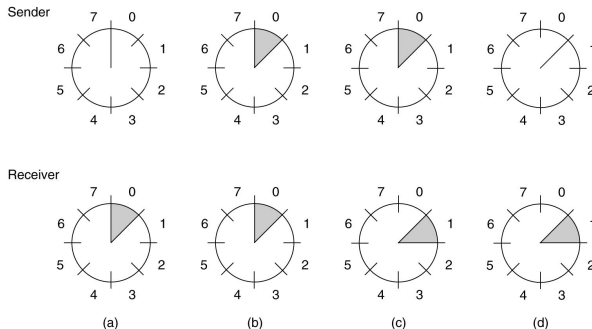


Figura: Ventana deslizante de tamaño 1, con un número de secuencia de 3 bits. (a) Al inicio. (b) Tras la transmisión de la primera trama. (c) Tras la recepción de la primera trama. (d) Tras recibir la primera confirmación de recepción

Protocolo de ventana deslizante de 1 bit

```
/* Protocol 4 (sliding window) is bidirectional. */

#define MAX_SEQ 1 /* must be 1 for protocol 4 */
typedef enum (frame_arrival, cksum_err, timeout) event_type;
#include "protocol.h"
void protocol4 (void)
{
    seq_nr next_frame_to_send; /* 0 or 1 only */
    seq_nr frame_expected; /* 0 or 1 only */
    frame r, s; /* scratch variables */
    packet buffer; /* current packet being sent */
    event_type event;

    next_frame_to_send = 0; /* next frame on the outbound stream */
    frame_expected = 0; /* frame expected next */
    from_network_layer(&buffer); /* fetch a packet from the network layer */
    s.info = buffer; /* prepare to send the initial frame */
    s.seq = next_frame_to_send; /* insert sequence number into frame */
    s.ack = 1 - frame_expected; /* piggybacked ack */
    to_physical_layer(&s); /* transmit the frame */
    start_timer(s.seq); /* start the timer running */

    while (true) {
        wait_for_event(&event); /* frame_arrival, cksum_err, or timeout */
        if (event == frame_arrival) { /* a frame has arrived undamaged. */
            from_physical_layer(&r); /* go get it */
            if (r.seq == frame_expected) /* handle inbound frame stream. */
                to_network_layer(&r.info); /* pass packet to network layer */
            inc(frame_expected); /* invert seq number expected next */
        }
        if (r.ack == next_frame_to_send) { /* handle outbound frame stream. */
            stop_timer(r.ack); /* turn the timer off */
            from_network_layer(&buffer); /* fetch new pkt from network layer */
            inc(next_frame_to_send); /* invert sender's sequence number */
        }
    }
    s.info = buffer; /* construct outbound frame */
    s.seq = next_frame_to_send; /* insert sequence number into it */
    s.ack = 1 - frame_expected; /* seq number of last received frame */
    to_physical_layer(&s); /* transmit a frame */
    start_timer(s.seq); /* start the timer running */
}
```

Protocolos con tamaño de ventana mayor de 1 bit

Si retardo de propagación mucho mayor que retardo de transmisión de la trama, el rendimiento del protocolo es muy pobre.

Ejemplo: $v_T = 50 \text{ kbps}$, $\text{Trama} = 1000 \text{ bits}$, $2 \times T_p = 500 \text{ msec}$

- ❶ $t = 0$ comienza envío de 1ª trama.
- ❷ $t = 20 \text{ msec}$ 1ª trama enviada.
- ❸ $t = 520 \text{ msec}$ llega confirmación de recepción.

Emisor transmitiendo durante $\frac{20}{520} = 4\% \rightarrow \text{Utilización red} = 4\%$

Protocolos con tamaño de ventana mayor de 1 bit

Solución: Emisor pueda enviar hasta w tramas antes de bloquearse, donde w igual al tiempo de propagación de ida y vuelta de la primera trama. Ej. anterior, $w = 26$

- 1 $t = 0$ comienza envío de 1ª trama.
- 2 $t = 20 \text{ msec}$ 1ª trama enviada, comienza envío segunda trama
- 3 $t = 520 \text{ msec}$ Final 26ª trama, llega confirmación de recepción de 1ª.

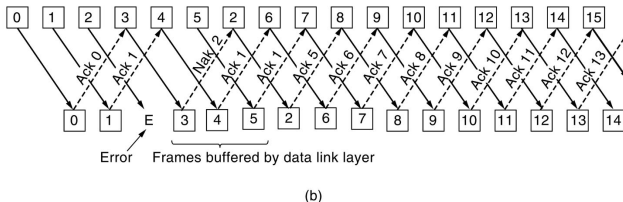
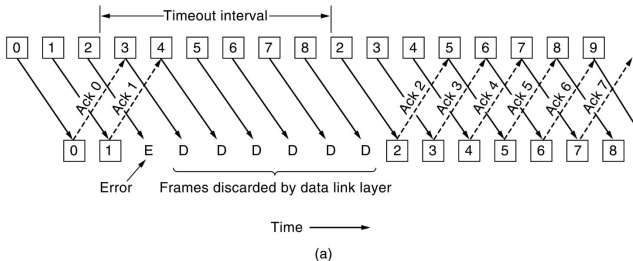
Emisor transmitiendo durante $\frac{520}{520} = 100 \%$

Protocolos con tamaño de ventana mayor de 1 bit

- Capacidad del canal es de b bits/seg.
- Tamaño de la trama de L bits.
- Retardo de propagación **de ida y vuelta** de R segundos.
- Si parada y espera, utilización de la línea = $L/(L + bR)$.
- Si $L < bR$, la eficiencia será menor que 50 %.

Canalización: Cuando $w > 1$

Variantes de canalización



Repetición selectiva: números de secuencia

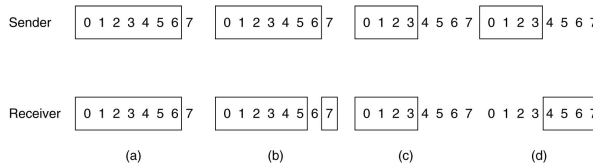


Figura: (a) Situación original con una ventana de tamaño 7. (b) Después de que se han enviado y recibido siete tramas, pero su recepción no se ha confirmado. (c) Situación inicial con un tamaño de ventana de 4. (d) Después de que se han enviado y recibido cuatro tramas, pero su recepción no se ha confirmado.

Contenido

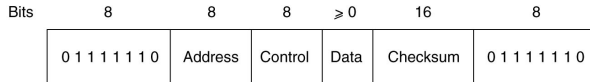
- 1 Introducción: problemática de acceso al medio
- 2 Protocolos de red de área local
 - CSMA persistente y no persistente
 - CSMA con detección de colisiones
 - Protocolos de reserva
- 3 Norma IEEE 802 para redes de área local
 - Norma IEEE 802.3 y Ethernet
 - Subcapa MAC de la IEEE 802.3z (Gigabit Ethernet)
 - NORMA IEEE 802.11 (Wifi)
- 4 La Subcapa de enlace
 - Parámetros de diseño de la capa de enlace
 - Detección de errores
 - Protocolos elementales de la capa de enlace
 - Protocolos de ventana deslizante
 - Ejemplos de protocolos de enlace de datos

HDLC-Control de Enlace de Datos de Alto Nivel

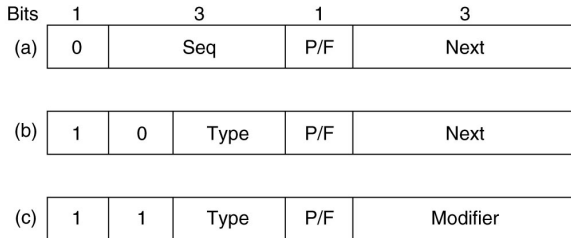
Protocolo	Nombre completo	Uso
HDLC	High level Data Link Control	Estándar ISO
ADCCP	Advanced Data Communications Control Procedure	Estándar ANSI
LLC	Logical Link Control	Estándar IEEE 802.2
LAP-B	Link Access Procedure Balanced	X.25
LAP-D	Link Access Procedure D-channel	RDSI (Señalización)
LAP-F	Link Access Procedure for Frame Mode Bearer Services	Frame Relay
LAP-M	Link Access Procedure - Modem	Módems RTC
PPP	Point to Point Protocol	Estándar Internet
SDLC	Synchronous Data Link Protocol	SNA (IBM)

Cuadro: Algunos miembros de la familia de protocolos HDLC

Estructura de la trama HDLC



Campo de control para distintos tipos de trama HDLC



Protocolo Punto a Punto (PPP): Usos

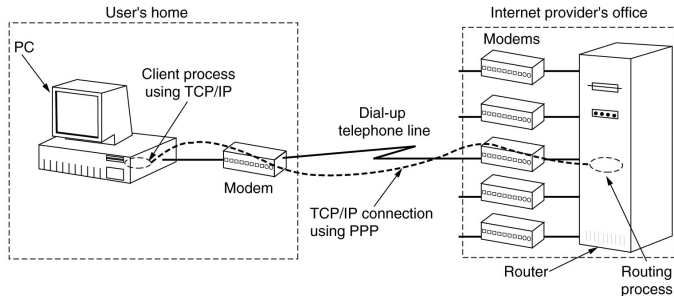


Figura: Computador personal doméstico que funciona como host de Internet.

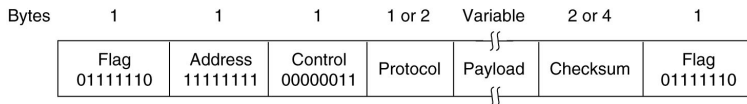
Protocolo Punto a Punto (PPP): Funciones

- Entramado y detección de errores.
- LCP (Protocolo de Control de Enlace): activar líneas, probarlas, negociar opciones y desactivarlas.
- NCP (Protocolo de Control de Red): negociar opciones distinta para cada protocolo de capa de red. soportado.

Protocolo Punto a Punto (PPP): Secuencia

- 1 PC llama al encaminador del proveedor (cable módem o módem ADSL).
- 2 Módem del encaminador establece una conexión física.
- 3 PC intercambia con encaminador paquetes LCP en el campo de carga útil de tramas PPP. Parámetros PPP a usar.
- 4 PC intercambia con encaminador paquetes NCP para configurar capa red (TCP/IP).
- 5 Cada proveedor de Internet asigna de manera dinámica direcciones IP a cada PC para su sesión.
- 6 El PC es ya un host de Internet y puede enviar y recibir paquetes IP

Protocolo Punto a Punto (PPP): Formato trama



Capa de Enlace en Frame Relay: LAPF

