

/* El protocolo 1 (utopía) considera la transmisión de datos en una sola dirección, del transmisor al receptor. Se supone que el canal de comunicación está libre de errores, y que el receptor es capaz de procesar toda la entrada a una rapidez infinita. En consecuencia, el transmisor se mantiene en un ciclo, enviando datos a la línea tan rápidamente como puede*/

```
typedef enum (frame_arrival) event_type;  
#include "protocol.h"
```

```
void sender1(void)  
{  
    frame s;                /*buffer para una trama de salida*/  
    packet buffer;          /*buffer para un paquete de salida*/  
  
    while(true){  
        from_network_layer(&buffer);  
        s.info = buffer;  
        to_physical_layer(&s);  
    }  
}
```

```
void receiver1(void)  
{  
    frame r;  
    event_type event;        /*no se ocupa aqui*/  
  
    while(true){  
        wait_for_event(&event);    /*la unica posibilidad es frame_arrival*/  
        from_physical_layer(&r);  
        to_network_layer(&r.info);  
    }  
}
```

/* El protocolo 2 (parada y espera) también contempla un flujo unidireccional de datos del transmisor al receptor. Se supone nuevamente que el canal de comunicación está libre de errores, como en el protocolo 1. Sin embargo, esta vez el receptor tiene capacidad finita de buffer y capacidad finita de proceso, por lo que el protocolo debe evitar explícitamente que el transmisor sature al receptor con datos a mayor velocidad de la que puede manejar*/

```
typedef enum (frame_arrival) event_type;  
#include "protocol.h"
```

```
void sender2(void)  
{  
    frame s;                /*buffer para una trama de salida*/  
    packet buffer;          /*buffer para un paquete de salida*/  
    event_type event;       /*frame_arrival es la unica posibilidad*/  
  
    while(true){  
        from_network_layer(&buffer);  
        s.info = buffer;  
        to_physical_layer(&s);  
        wait_for_event(&event);  
    }  
}
```

```
void receiver2(void)  
{  
    frame r, s;  
    event_type event;  
  
    while(true){  
        wait_for_event(&event);  
        from_physical_layer(&r);  
        to_network_layer(&r.info);  
        to_physical_layer(&s);  
    }  
}
```