

Universidad De Los Andes  
Ingeniería y Ciencias Aplicadas  
Sistemas Operativos y Redes

## Tarea 3 Parte II

Diseño del protocolo para la aplicación.  
(Modificaciones)

## Juan Esteban Rodríguez Aguayo.

### Descripción:

Esta tarea tiene como objetivo la comunicación entre dos o más usuarios mediante Internet, teniendo la capacidad de poder enviar archivos y mensajes, además de poder crear grupos de conversación entre distintos usuarios. La aplicación deberá ser capaz de recibir mensajes en forma paralela a enviarlos, así no queda en estado “waiting for message”, o “esperando un mensaje”, también tendrá el deber de guardar la conversación con cada usuario, mostrando su estado para poder realizar una comunicación entre las partes, si y solo si ambas están en estado conectado, dando paso a ser un servicio no centralizado.

Para esta tarea, se utilizará el protocolo de datagrama TCP, y una mezcla de los protocolos client-server, donde toda la información será entregada al servidor, y este a quien corresponda. La aplicación tendrá dos tipos de comunicación entre usuarios, personal, y en grupo. Para lograrlo cada uno de los tipos de comunicación usa protocolos distintos, donde el servidor distingue por la cabecera del mensaje la finalidad de este, ambos utilizan el protocolo de conexión a host TCP.

### Tipo de Host:

Los hosts serán de tipo peer-to-peer para las comunicaciones entre dos usuarios, para las comunicaciones en grupo, o varios usuarios, se utilizará una mezcla de ambos protocolos. Para enviar los mensajes, el servidor actuará de server o HUB, enviando el mensaje a todos los hosts que se encuentren en el grupo, y gracias al protocolo TCP, es posible para el emisor de mensaje ver a quienes aún no le llega el mensaje (funcionalidad tipo whatsapp para ver si el mensaje fue recibido).

Al crear un grupo, cada dispositivo intentará abrir un puerto para enviar y recibir con cada dispositivo, así al enviar, se envía todos os puertos de salida, pero al escuchar llega según cada usuario envíe mensajes.

### Formato de Mensaje:

Puertos de origen y destino, 16 bytes para cada uno, cada uno con su número correspondiente.

Numero de secuencia, 4 bytes, indica el número de paquete.

Número de acuso de recibo, 4 bytes, indica que paquete fue recibido.

Data offset, 4 bits, indica el tamaño de palabras de 32 bits de la data.

Bits de Control, 6 bits, indica si el mensaje es urgente, si es acuso de recibo, si se debe ingresar la data se ingrese inmediatamente a la aplicación, resetear la conexión en caso de pérdida o problemas relacionados, y finalizar la conexión.

Ventana, 2 bytes, indica cual es el tamaño el buffer que el receptor está dispuesto a aceptar.

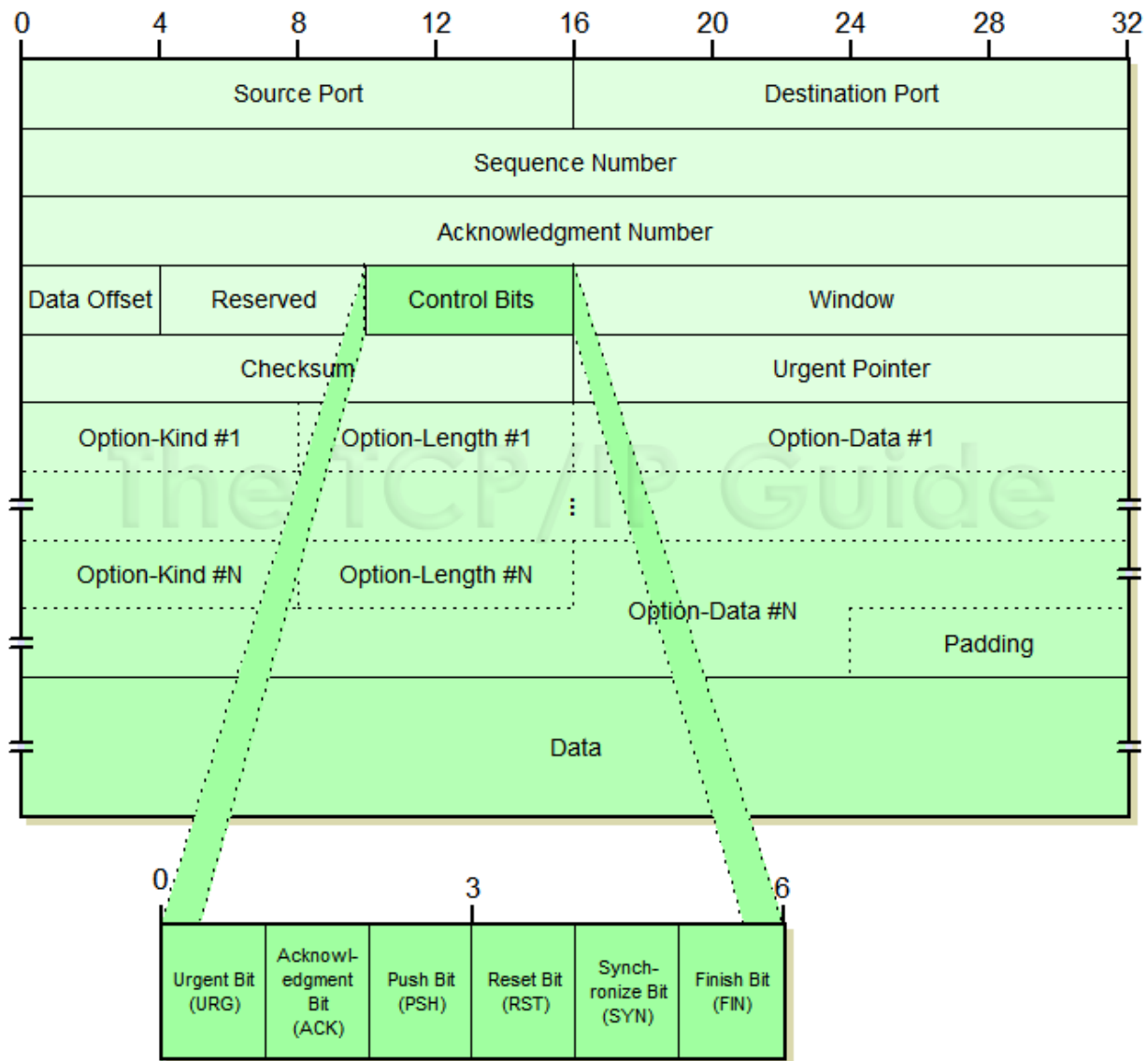
Checksum, 2 bytes, utilizado para comprobar la integridad del mensaje.

Puntero Urgente, 2 bytes, junto al bit de control es utilizado para enviar data con prioridad con respecto al resto.

Opciones, 2-8 bits, será utilizado para distinguir entre envío de mensaje o archivo, y posibles estados y/o información relevante.

Padding, llena con ceros, si Opciones no es de 32 bits.

Data, información que se desea enviar.



(Fuente: [http://www.tcpguide.com/free/t\\_TCPMessageSegmentFormat-3.htm](http://www.tcpguide.com/free/t_TCPMessageSegmentFormat-3.htm))

El tamaño de cada paquete se definirá al ir implementando la segunda parte de la tarea.

## Comunicación:

Se utilizarán datagramas confiables, con protocolo TCP, para que la información llegue integra, ya que, los archivos además de ser de tamaños variables, si no están íntegros la probabilidad de que un archivo grande (>10 Mb) no llegue integro crece con el tamaño del archivo.

## Interacciones:

