

# Protocolos de Transporte: UDP, TCP

(parte I)

Andres Barbieri

barbieri(at)cespi.unlp.edu.ar

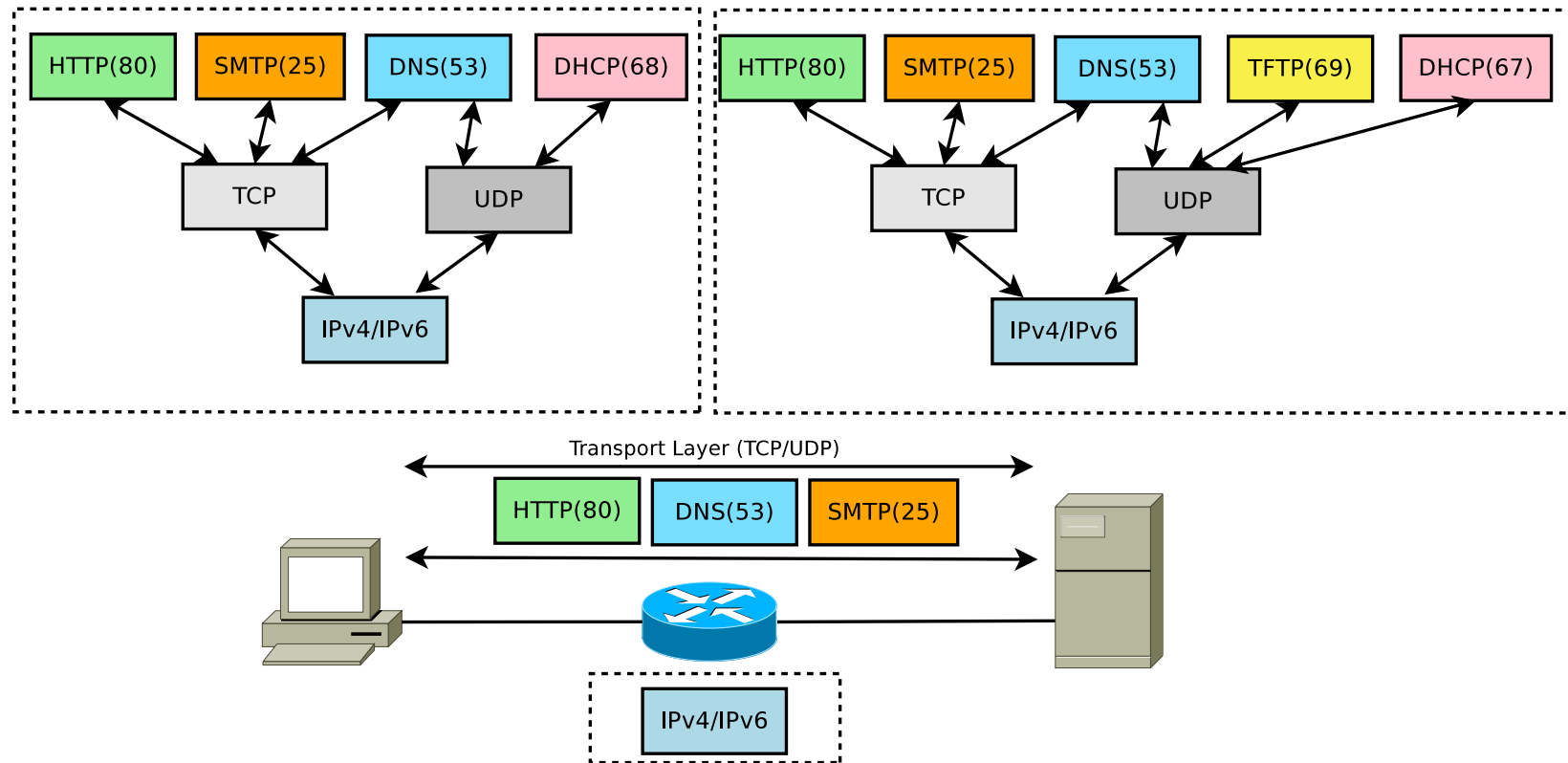


## Introducción

- IP provee un servicio “débil”, pero eficiente (best-effort).
- Para IP los paquetes pueden ser descartados, des-ordenados, retardados duplicados o corrompidos.
- Paquetes IP solo dirección DST y SRC, ¿Como elegir la App.?
- ¿Cómo enviar info sobre la red de acuerdo al estado de la misma?

## Características de Transporte

- MUX/DEMUX App. to App. (Ports).
- Soporte de datos de tamaños arbitrarios.
- Control de Errores.
- ¿Cuándo y Cómo una App. debe enviar datos?
  - Control de Flujo.
  - Control de Congestión.
- Dos modelos:
  - Modelo Confiable: TCP.
  - Modelo NO Confiable: UDP.



## UDP

- User Datagram Protocol (RFC-768).
- Protocolo Minimalista. Menor Overhead.
- Características de IP: best-effort.
- Orientado a Packets/Datagramas.
- PDU: Datagrama (Por coherencia con nivel Transporte se o suele llamar Segmento).
- Solo provee MUX/DEMUX.
- No incrementa Overhead end-to-end.
- No requiere establecimiento de conexión.
- Servicio FDX.
- Aplicaciones video/voz streaming/TFTP/DNS/Bcast/Mcast.

## TCP

- Transport Control Protocol (RFC-793)
- Protocolo confiable, ordenado, buffering, control de flujo y de congestión.
- Orientado a Streams.
- PDU: Segmento.
- Provee MUX/DEMUX.
- Incrementa Overhead end-to-end a costa de confiabilidad.
- Requiere establecimiento de conexión (y cierre).
- Servicio FDX.
- Aplicaciones FTP/HTTP/SMTP/acceso remoto/Unicast.

## Headers/Encabezados

- El encabezado IP provee: Ruteo, Fragmentación, Detección de algunos errores.
- El encabezado UDP provee: MUX/DEMUX, Detección de errores (no obligatorio).
- El encabezado TCP provee: MUX/DEMUX, Detección de errores, Sesiones, Control de Flujo y Control de Congestión.

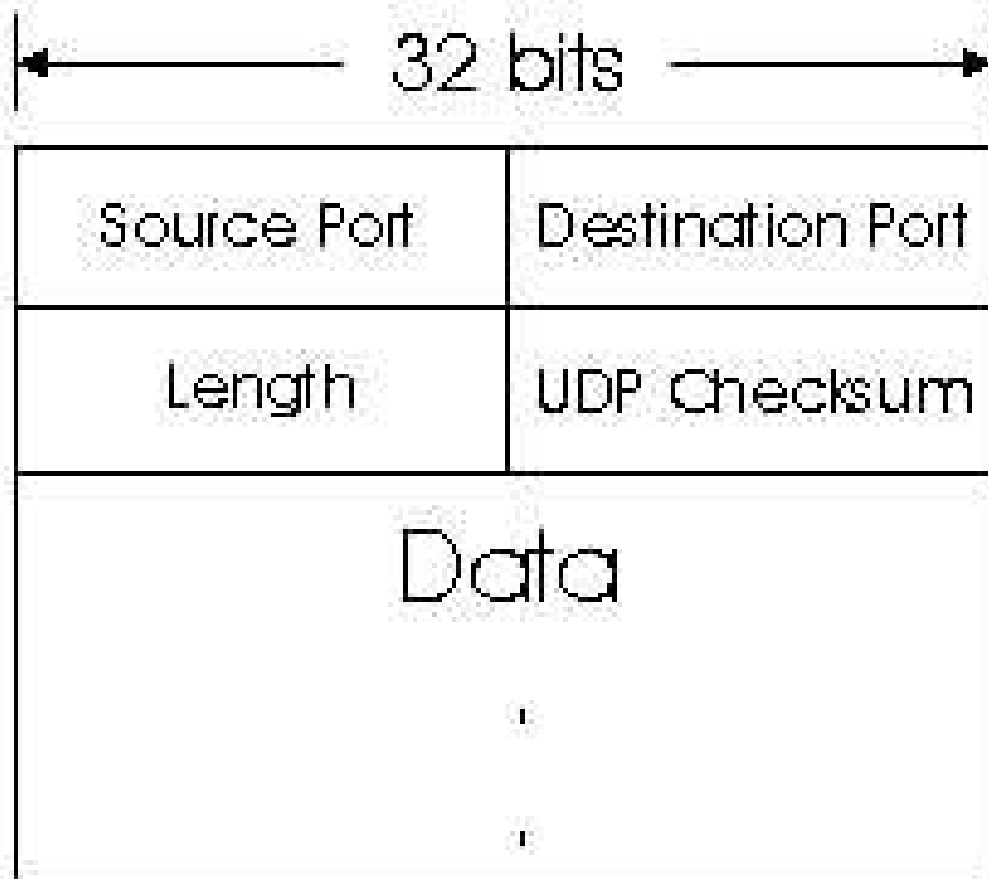
```
? grep udp /etc/protocols
```

```
udp      17      UDP # user datagram protocol
```

```
? grep tcp /etc/protocols
```

```
tcp      6      TCP # transmission control protocol
```

## Datagrama UDP

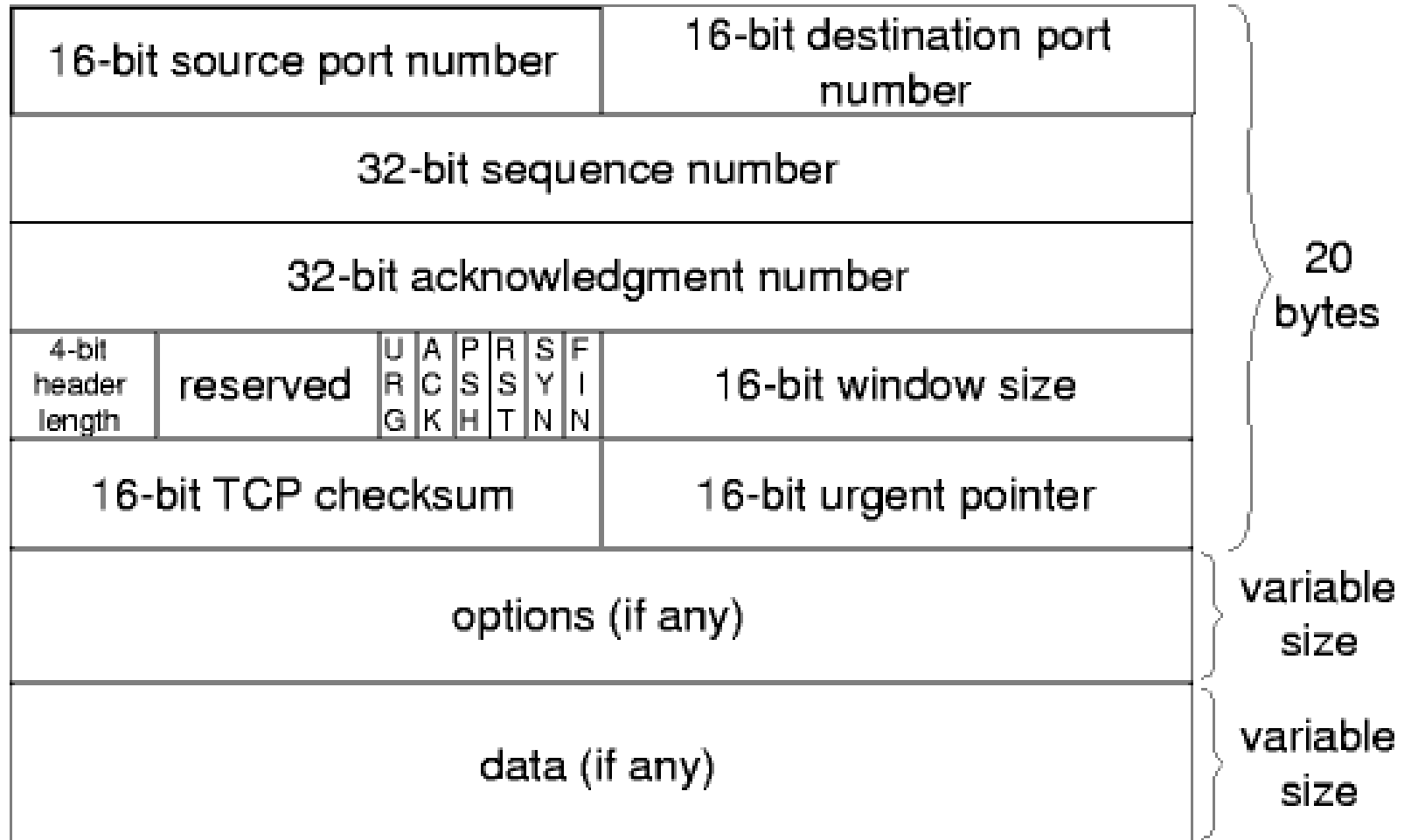




## Datagrama (UDP)

- Puertos: MUX/DEMUX.
- Longitud: UDP HDR + Payload.
- Checksum
  - Cálculo Ca1, Opcional. 0 = Sin checksum.
  - Calculado HDR + PseudoHDR + Payload.
  - PseudoHDR: IP.SRC + IP.DST + Zero + IP.Proto + UDP.LENGTH.
  - PseudoHDR: protección contra paquetes mal enrutados.
  - Aplicaciones de LAN por eficiencia lo podrían deshabilitar.
  - Si tiene error se descarta silenciosamente.

## Segmento TCP



## Segmento (TCP)

- Puertos: MUX/DEMUX.
- No tiene Longitud total, si de HDR LEN (variable, max 60B Unit=4B).
- Total LEN se computa para PseudoHDR, no viaja en el segmento.
- Checksum:
  - Cálculo Ca1, Obligatorio.
  - Calculado de forma igual que UDP.
  - Si tiene error podría pedir retransmisión, implementación de TCP descarta y espera TMOUT.
- Necesidad de manejar Timers, TMOUT por cada segmento. (implementaciones lo manejan más eficiente).

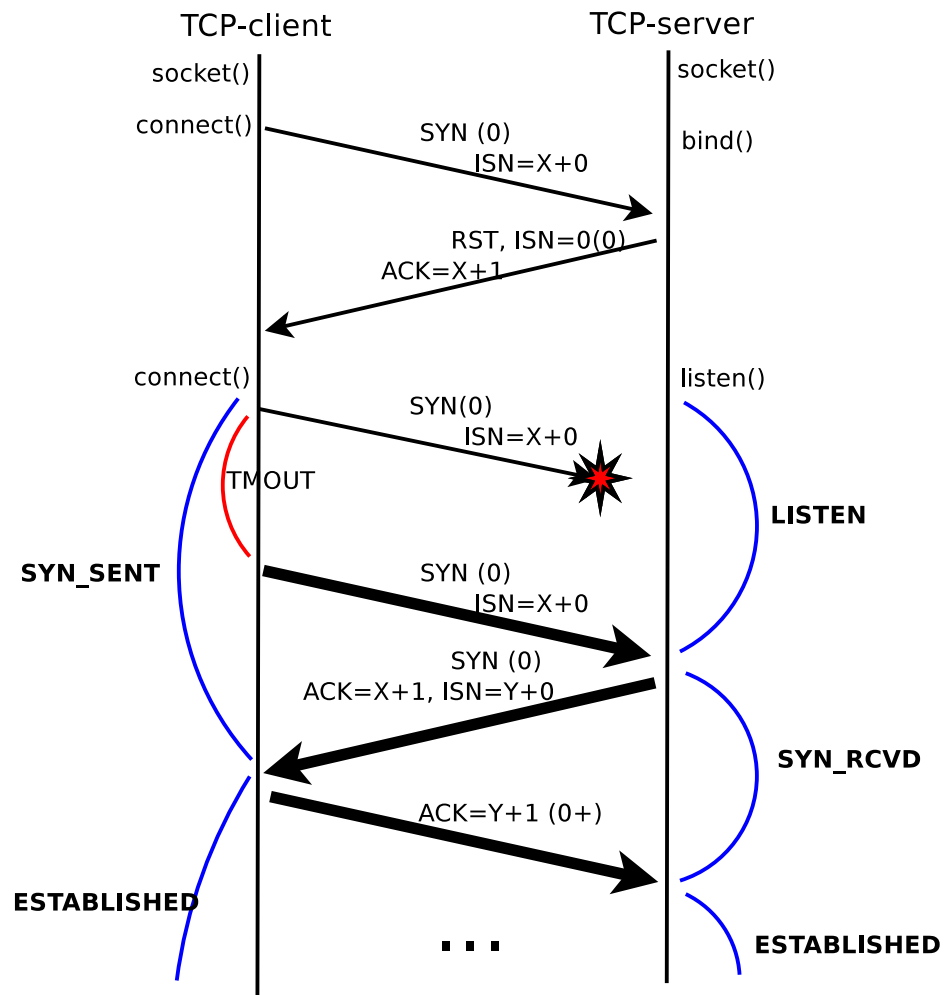
## Segmento (TCP) (Cont'd)

- Campos de Sesiones: FLAGS: Syn, Fin.
- Máquina de estado finita por cada conexión.
- Campos de Control de Flujo: ACK, Seq, Ack Seq, Win.
- Permite Opciones y Negociación.
- TCP entrega y envía los datos agrupados o separados de forma dis-asociada de la aplicación:
  - La aplicación puede enviar 300 bytes en un write y TCP lo podría enviar en 3 segmentos separados de 100 bytes c/u.
  - La aplicación puede enviar 100 bytes y luego otros 200 y TCP esperar para enviarlos todos juntos.
  - La aplicación puede intentar leer 200 bytes del buffer y TCP solo entregar 150 bytes y luego el resto.

## TCP Establecimiento de Conexión

- 3Way-Handshake (3WH).
- En el 3 segmento se puede enviar info.
- el ISN debe ser un contado que se incrementa cada 4 mseg.
- RST si no hay proceso en estado LISTEN.
- Open Pasivo y Activo.
- Open simultáneo.

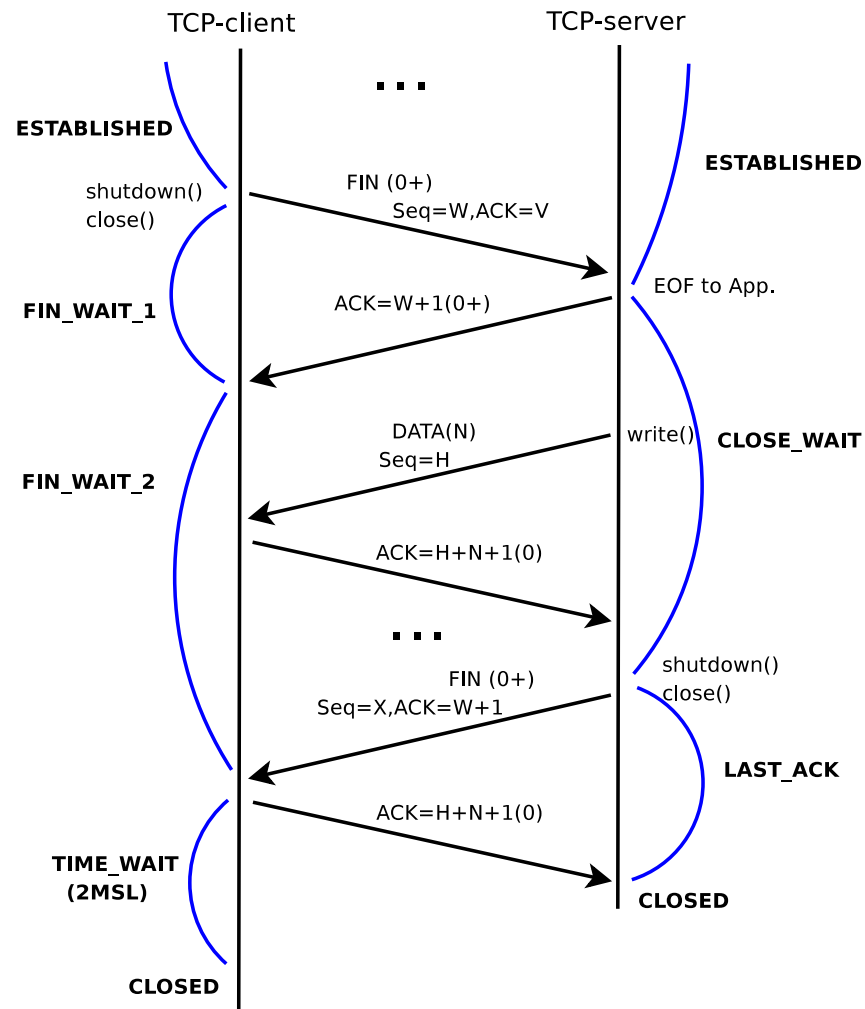
## 3 Way Handshake



## TCP Cierre de Conexión

- 4Way-Close (4WC).
- Posibilidad de Half-Close.
- Podría cerrarse en 3WC.
- Espera en TIME\_WAIT, 2MSL (aprox. 2\*2min).
- Evitar con SO\_REUSEADDR.
- Cierre incorrecto con RST.
- Close simultáneo.

# TCP Close

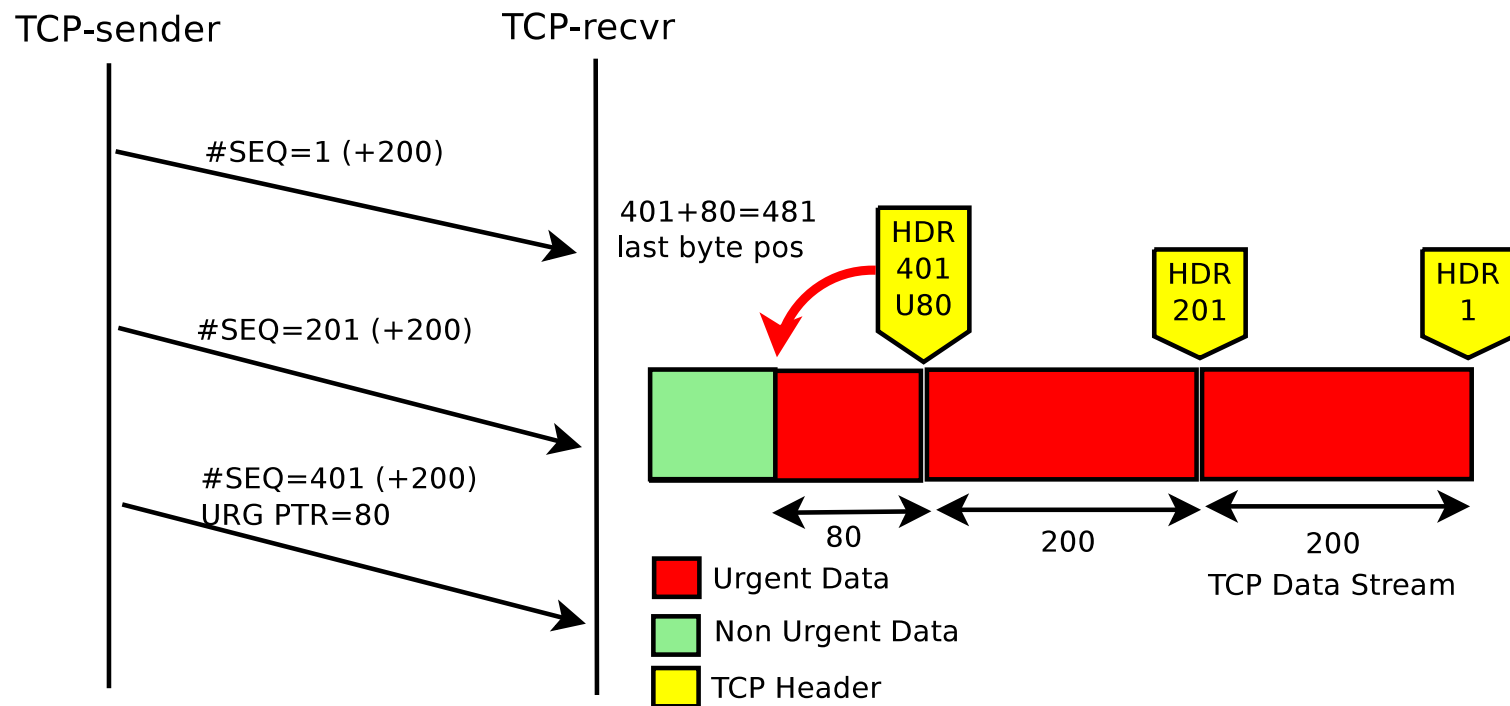




## Otros campos TCP

- TCP entrega y envía los datos agrupados o separados de forma dis-asociada de la aplicación.
- Datos Urgentes: URG.
  - Urgent Pointer válido si URG=1.
  - Indica:  $\text{offset positivo} + \text{Seq Num} = \text{last Data Urgent byte}$ .
  - Indicar a la App. datos urgentes, debe leer.
  - Debería combinarse con PSH. Habitualmente llamado OOB data (TCP no soporta OOB!!!).
- Pushear datos: PSH.
  - Fuerza a TCP a pasar datos a la App.
  - No lo deja "Bufferear" los datos recibidos (Input).

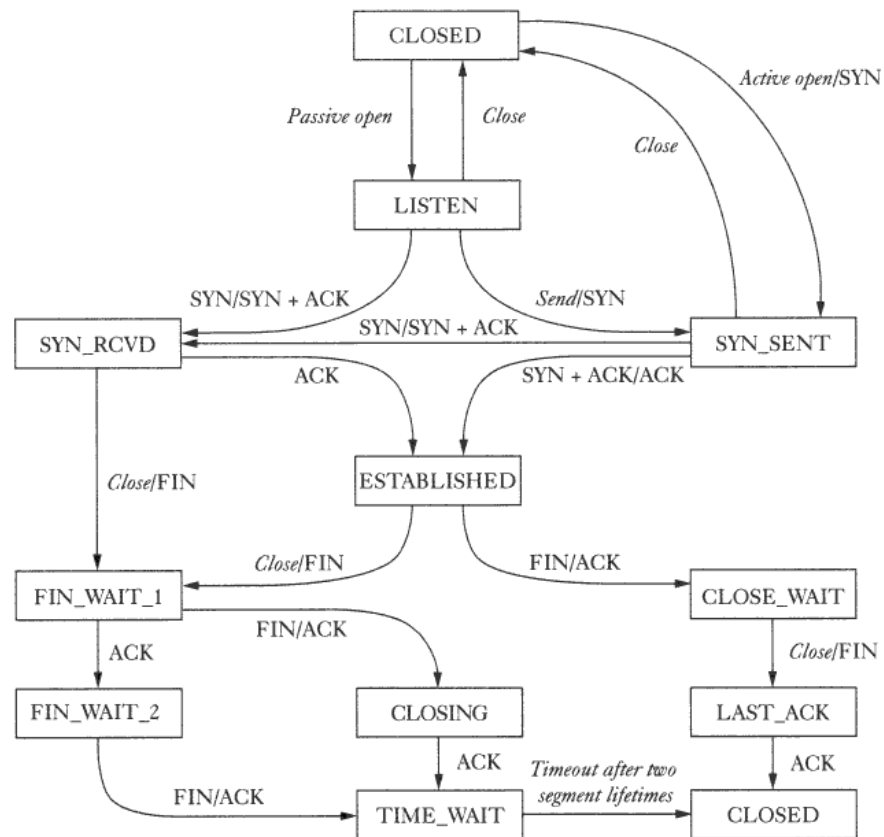
## Urgent Pointer



## Opciones TCP

- Maximum Segment Size (MSS), recomendado 536B, RFC-879.
- Window Scaling.
- Selective Acknowledgements (SACK).
- Timestamps.
- NOP.

# TCP Diagrama de Estados



## Fuentes de Información

- Kurose/Ross: Computer Networking (5th Edition).
- TCP/IP Illustrated, Volume 1: The Protocols, W. Richard Stevens.
- RFCs: <http://www.faqs.org/rfcs/rfc793>, rfc798, ...
- Wikipedia <http://www.wikipedia.org>.
- Slides de la Prof. Paula Venosa.
- TCP/IP Guide: <http://www.tcpipguide.com/>.
- Internet ...