Nuevos protocolos MPTCP

Área de Ingeniería Telemática
Dpto. Automática y Computación
http://www.tlm.unavarra.es/

Remember TCP

TCP

- Protocolo de transporte fiable sobre IP
- Proporciona
  - Conexiones
  - Transporte fiable
  - Control de flujo
  - Control de congestion

TCP

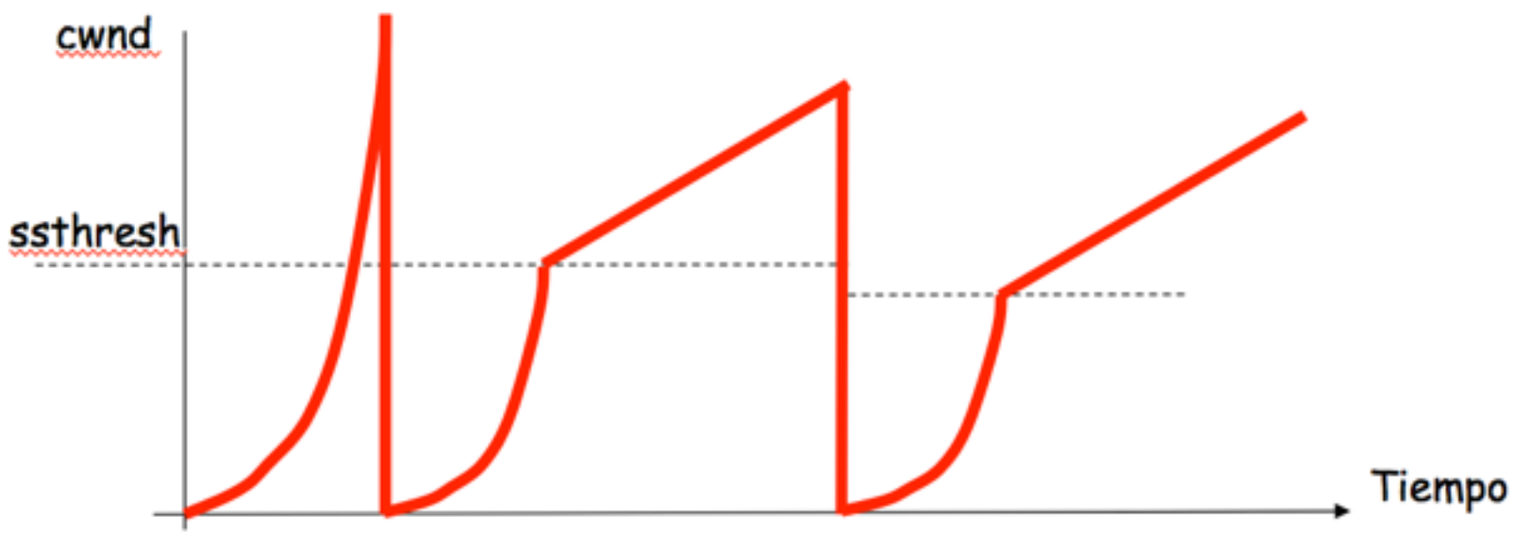
- Conexiones
- Identificadas por la 4 tupla

  [IPlocal, puertolocal, IPremota, puertoremoto]

- Ligada al interfaz
  - Un host al cambiar de red no puede reutilizar la conexión
  - Un movil con wifi y 3G no puede usar la misma conexión en los dos
  - Un servidor con multihoming no puede usar la misma conexión si se cae uno de sus interfaces

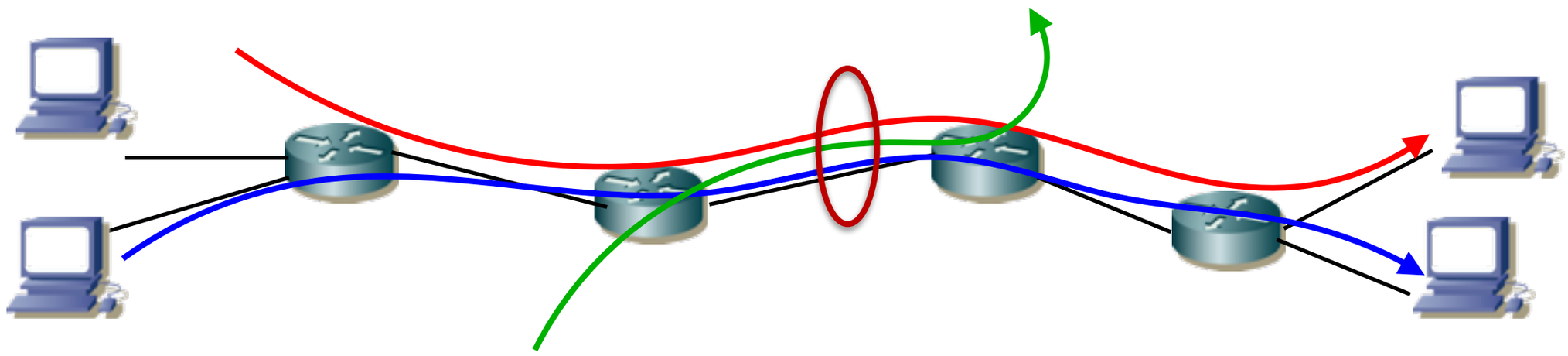
TCP

- Control de congestion
  - Ventana de congestion
  - Básico AIMD
    - si no hay perdidas la ventana sube poco a poco (lineal)
    - si hay perdidas la ventana baja agresivamente (a 0 o 0.5w)
  - Slow start al comienzo
  - Mecanismos avanzados: fast-recovert, fast-retransmit…



TCP

- Congestion y fairness
  - Las conexiones que comparten enlace tienden a la larga a repartirse el ancho de banda de forma justa



Problemas multiples caminos…

- Servidor con varias direcciones IP
  - Una conexión TCP solo me permite comunicarme con una
  - Se podría aprovechar a nivel de aplicación (pero require aplicación que lo soporte)
- Cliente con varias direcciones IP
  - Tengo que elegir cual uso al conectarse
  - Si tiene varias conexiones de red no puede usarlas para redundancia. La conexión TCP se perderá cuando caiga su dirección IP
  - Se podría hacer a nivel de aplicación

- No podría hacer todo esto TCP por su cuenta?

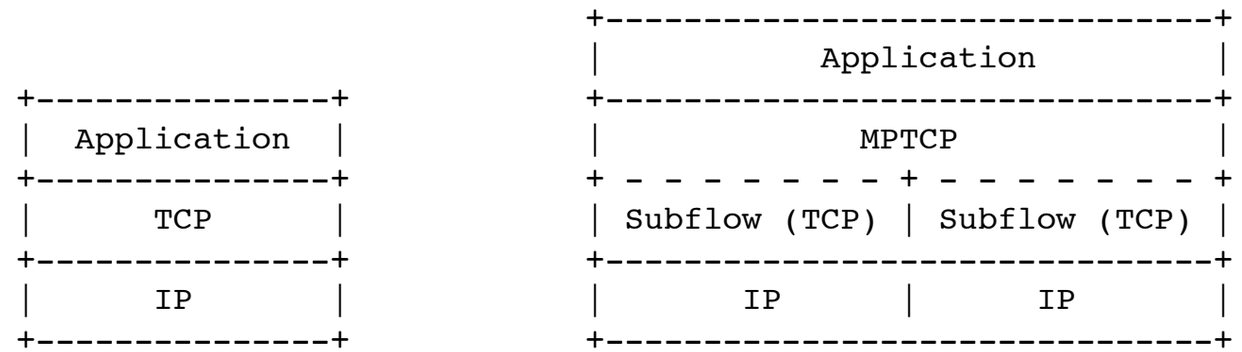
Multi Path TCP (MPTCP)

MPTCP

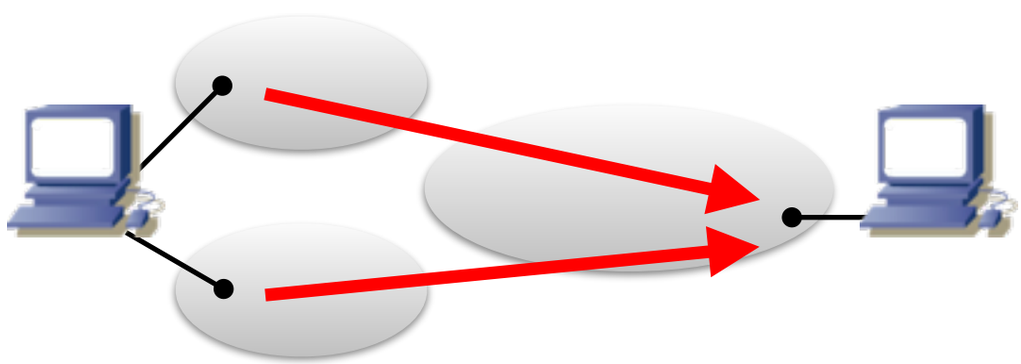
- IETF WG(2009) RFC 6824 (Ene 2013)
- Multi Path TCP
- Idea
  - Permitir Inverse Multiplexing de una sesión TCP sobre varios flujos TCP
  - Que permita utilizar el ancho de banda de varios canales
  - Que permita utilizar varios canales como alternativos y sobrevivir a caidas o cambio de IP
  - Que presente un interfaz TCP a las aplicaciones (Transparente)
  - Las aplicaciones existentes puedan usarlo sin modificarse
  - Que se degrade a TCP normal si algun extremo no lo soporta

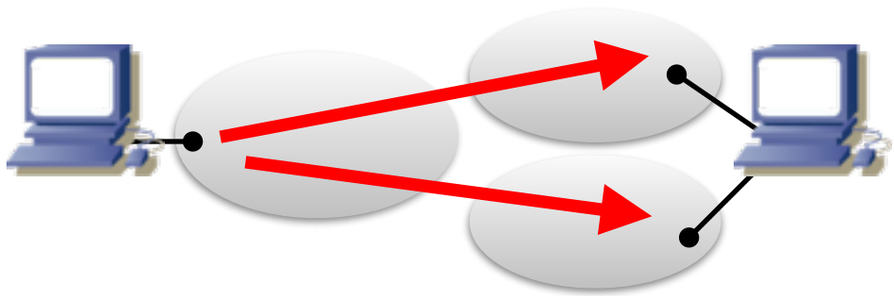
MPTCP idea

- Una sesión MPTCP se transporta por varios subflujos que son conexiones TCP



- Se pueden añadir (o eliminar) dinámicamente flujos
- Se puede suponer que uno de los dos extremos es multihomed





Manejo de subflujos

Inicio

- Conexión TCP 3 way handshake
- TCP option MP_CAPABLE [incluye key]
  - key de 64 bits generada por cada endpoint
  - identifican la conexión (endpoint)
- El otro extremo responde con MP_CAPABLE y su key o ignora
- ACK repite las dos keys

```
Host A                                          Host B
------                                          ------
MP_CAPABLE                  ->
[A's key, flags]
                            <-                  MP_CAPABLE
                                                [B's key, flags]
ACK + MP_CAPABLE            ->
[A's key, B's key, flags]
```

- Al añadir subflows se usara un token derivado de la key
- Flags para checksum opcional, elegir cryptografia y extensiones…

Añadir subflow

```
Host A                                  Host B
------                                  ------
MP_JOIN                 ->
[B's token, A's nonce,
 A's Address ID, flags]
                        <-              MP_JOIN
                                        [B's HMAC, B's nonce,
                                         B's Address ID, flags]

ACK + MP_JOIN           ->
[A's HMAC]

                        <-              ACK
```

- Token=hash(Key). Es el indice de la tabla de conexiones
- nonce para autentificar
- HMAC de B depende de token y nonce A
- El Address ID identifica a una dirección dentro de la conexión (independientemente de NATs intermedios)

- El subflow se añade a la conexión indicada

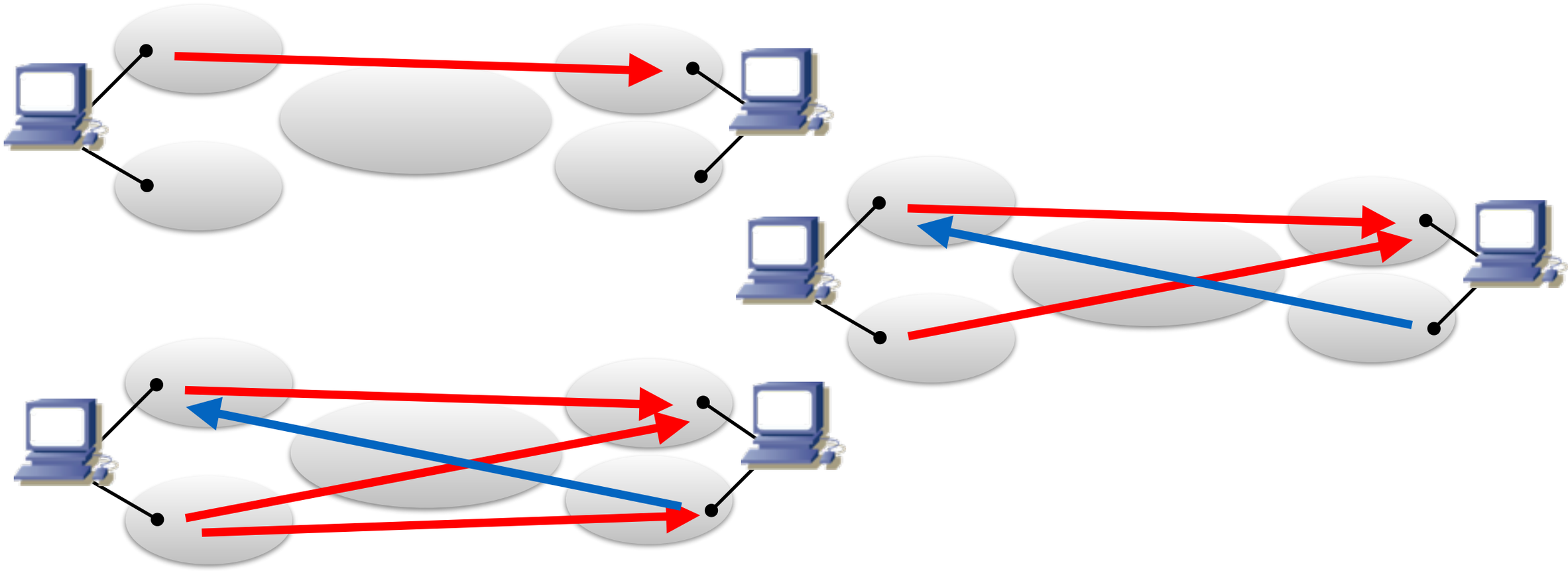
Informar de dirección

- ADD_ADDR Direccion, Addressid, [puerto]
  - Puerto opcional, por defecto se usa el mismo en todas las direcciones
  - Notifica al otro extremo que también tienes esta dirección asignándole un addressid
  - Se soporta IPv4 e IPv6

- REMOVE_ADDR Addressid
  - Una dirección no es valida (interfaz caído por ejemplo)
  - El otro extremo puede cerrar subflows
  - Debe hacer keepalive
  - Si falla debe hacer RST del TCP subflow

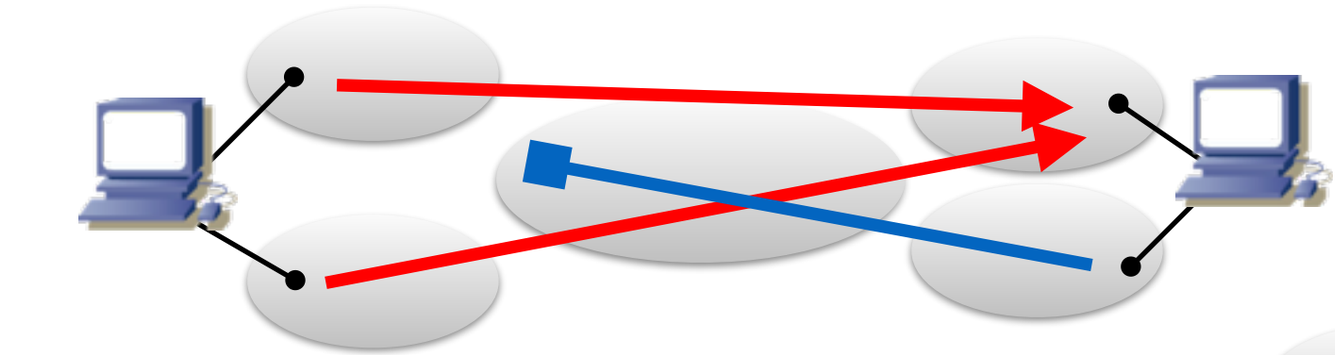
Dos métodos para conocer direcciones

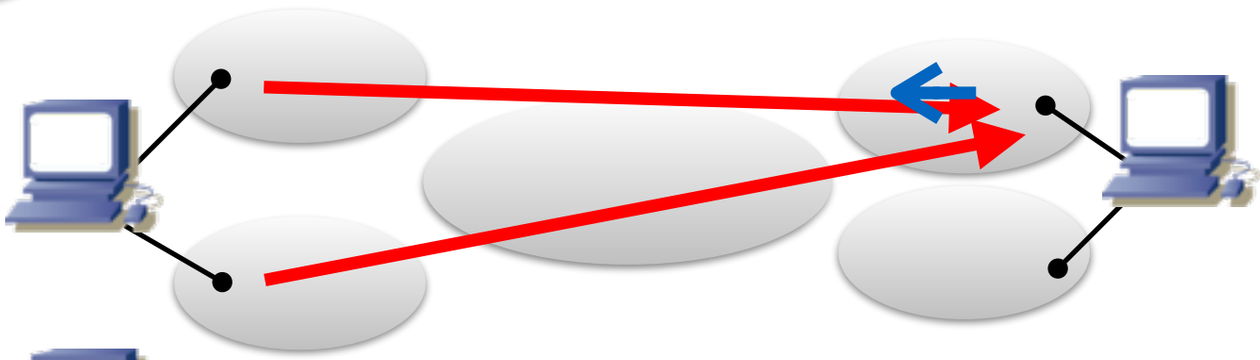
- A inicia conexión con MP_CAPABLE
- Si el otro extremo responde MP_CAPABLE
- A sabe que tiene varias IPs luego inicia nuevos conexiones con MP_JOIN
- B también sabe que tiene varias IPs puede iniciar nuevas conexiones con MP_JOIN a la dirección de A que conoce
- **Los MP_JOINS permiten conocer direcciones del otro lado**

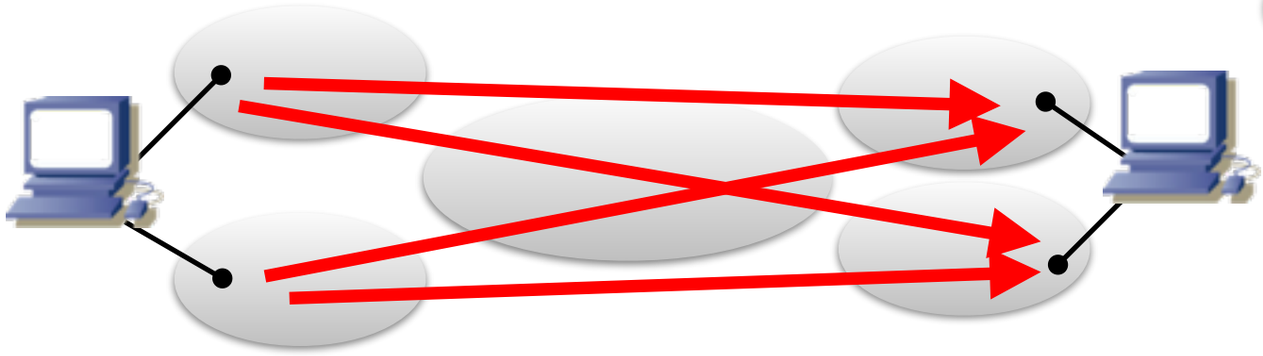


Dos métodos para conocer direcciones

- A inicia conexión con MP_CAPABLE y nuevos flujos con MP_JOIN
- Los MP_JOINs de B a la dirección de A no llegan (por un NAT por ejemplo)
- B informa de sus otras direcciones con ADD_ADDR en los flujos que tiene con A
- A conoce nuevas direcciones de B así que inicia subflujos con MP_JOIN

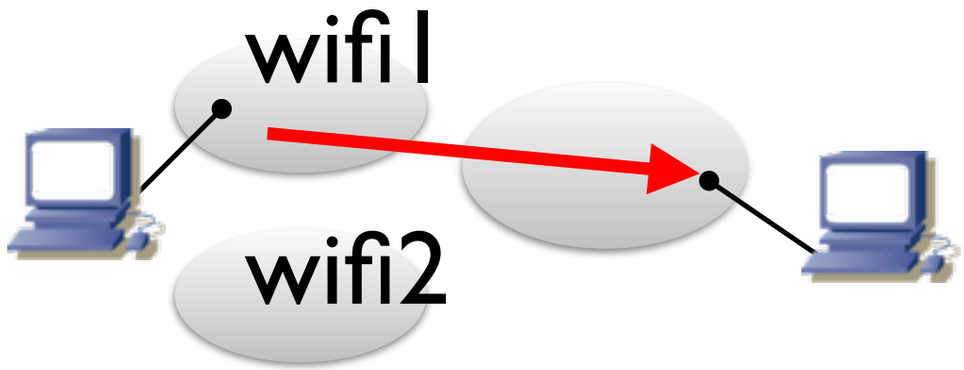


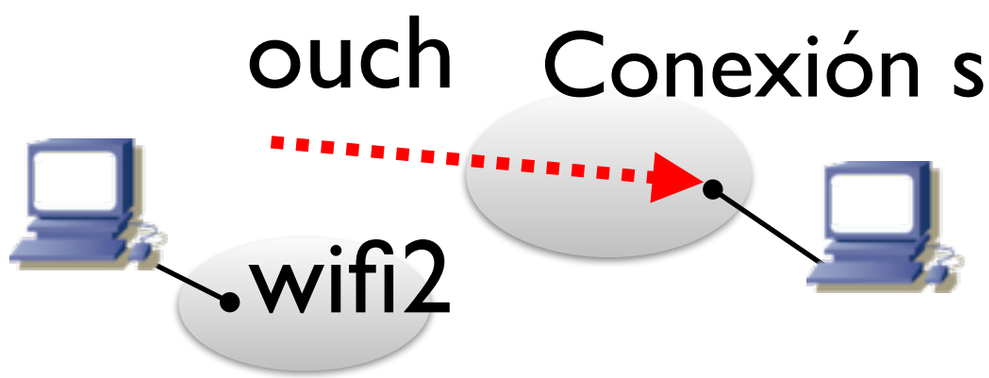




Path management

- Ambos extremos intentan mantener flujos entre todas las combinaciones de direcciones dinámicamente
- Hay cierre de conexión global
- Cerrar todos los flujos no implica el cierre inmediato de la conexión sino tras un timeout
- Para permitir break-before-make handover





La conexión no se interrumpe

Data transfer

Data transfer

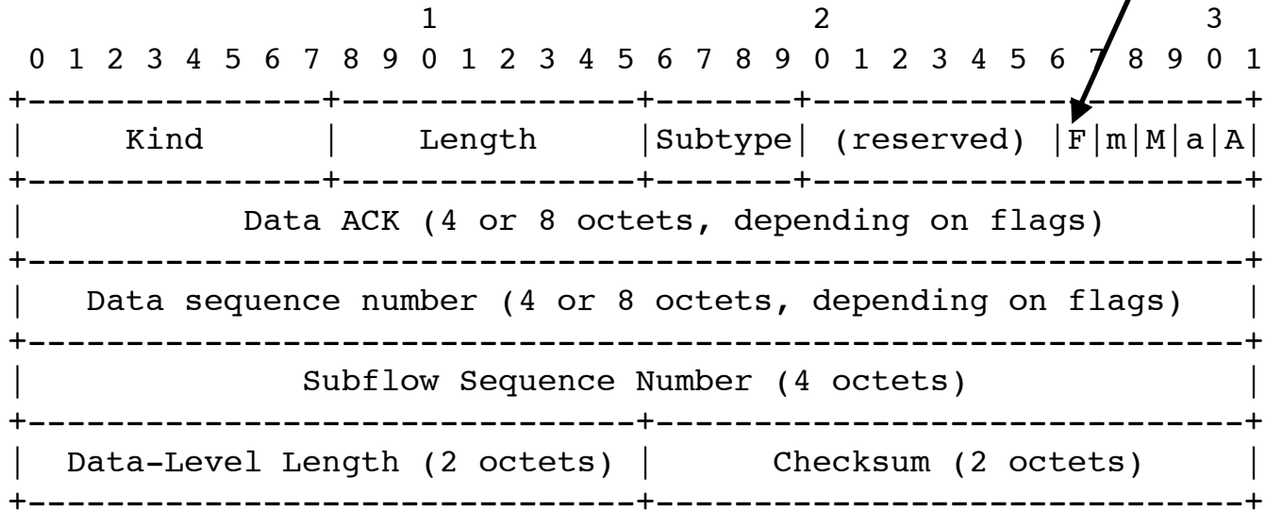
- Los datos de aplicación (entregados al socket) se envían por los diferentes subflows
- Con suficiente información para ser reensamblados en destino

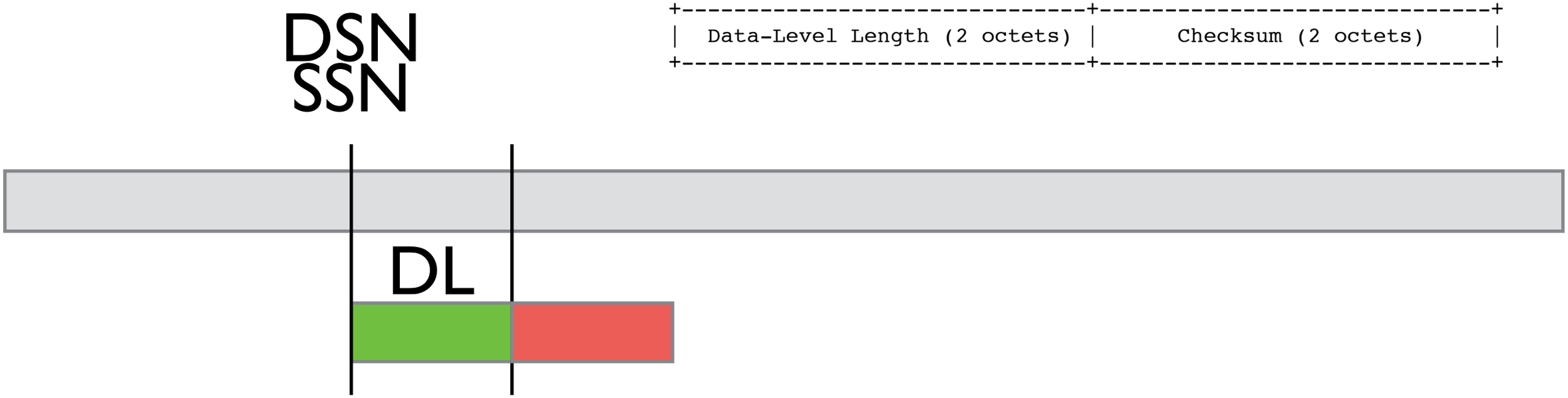
- Global Data Sequence Number (64 bits)
- El de TCP era de 32bits

Data transfer

- Mapeo del seq de subflow al global
- 64bits Data Sequence Number (global) DSN
- Opcion TCP: DATA_SEQUENCE_SIGNAL DSS

DATA FIN





- El mapeo no tiene por que enviarse en cada paquete de datos

Data transfer y subflows

- ACK
  - Los datos de subflow se confirman aunque queden huecos a nivel global
  - Para evitar interbloqueos

- Flow control
  - La ventana anunciada se interpreta como permitda desde el Data ACK por cualquiera de los paths

Retransmision

- Se puede retransmitir en diferente subflow
- Pero hay que retransmitir también en el subflow original para no corromper el numero de secuencia del subflow

Congestion control

- El RFC no propone un algoritmo claro
- Hay uno definido en RFC 6356 (2011)
- Pero pueden aparecer mas

Cerrar MPTCP conexión

- DATA_SEQUENCE_SIGNAL [data fin]
- Debe ser confirmada
- Mecanismo parecido al 4-way FIN de TCP
- Cierre global de la conexión

- De lo contrario sigue activa aunque se cierren todos los flujos
- Si se cierran todos los flujos puede acabar por timeout

- También hay MP_FASTCLOSE equivalente a TCP RST
- Y MP_FAIL para volver al modo no-MP en caso de problemas con middleboxes

Prioridad

- Un flujo al iniciarse se establece como Regular o Backup
  - Con un flag en MP_JOIN
- Los backup solo se usan si no hay regulares operativos
- Se puede cambiar la prioridad de un flujo con MP_PRIO

Control

Control desde el sistema operativo

- Path-manager
  - Via sysctl varios disponibles
    - default : no inicia subflujos pero los acepta
    - fullmesh : intenta tener todas las combinaciones
    - ndiffports : todas las combinaciones con varios puertos por combinación
    - …
- Scheduler (por donde envío)
  - Via sysctl
    - default : envia por los que tienen menos RTT primero
    - roundrobin : reparte por todos
    - …

Control desde las aplicaciones

- Pueden elegir con opciones del socket que se inicien o no subflujos
- RFC 6897 Application Interface considerations

MPTCP status

Implementations

- Linux
- iOS7+ (Lo usa Siri)
- Citrix NetScaler
- …

Usos

- Mantener conexión en móviles, cuando los proveedores 3g o wifi desaparecen
- Balanceo de carga en escenarios con varios caminos/ proveedores
- Fiabilidad en escenarios con varios caminos/proveedores

- Handoff en wifis? pero solo si puedes tener IP en dos simultáneamente
- Migración de maquinas virtuales entre datacenters con diferentes IPs