



DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE



COMUNICACIÓN

13169

SISTEMAS DISTRIBUIDOS



Comunicación

- La comunicación entre procesos se encuentra en el núcleo de los SD.
- Necesidad de analizar las formas en que los procesos desarrollados en distintas máquinas pueden intercambiar información.
- La comunicación por paso de mensajes no es trivial y es más compleja que las primitivas por manejo de memoria compartida.
 - Redes de comunicación poco confiables, como internet.
 - Distintos SO y sistemas de archivos.
 - Saturación de la red y latencia.



Fundamentos de Comunicación

Analicemos los protocolos de comunicación de redes.

Protocolo en Capas

- No existe memoria compartida a nivel de SD, por lo que toda la comunicación se realiza a través del envío y recepción de mensajes a bajo nivel.
- Si el proceso A debe comunicarse con el proceso B, primero elabora un mensaje en su propio espacio de dirección.
- Luego, ejecuta una llamada al SO el cual envía el mensaje sobre la red hacia B.
- Si bien parece sencillo, debe existir un acuerdo entre los procesos sobre el significado de los bits, largos de la trama, acciones en caso de error, entre otros.



Fundamentos de Comunicación

Protocolo en Capas: Modelo OSI

- Facilita el manejo de varios niveles y consideraciones de la comunicación.
- Diseñado para permitir que los sistemas abiertos se comuniquen.
- Un sistema abierto es aquel que está preparado para comunicarse con cualquier otro sistema abierto mediante reglas estándar que regulen el mensaje.
- Estas reglas son lo que se denomina **protocolos**.
- **Protocolos orientados a conexión.**
- **Protocolos no orientados a conexión.**

Fundamentos de Comunicación

Protocolo en Capas: Modelo OSI

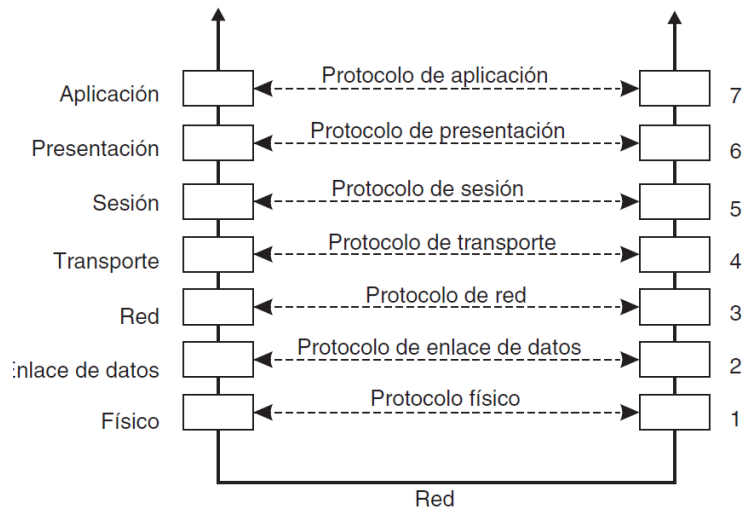


Figura 4-1. Capas, interfaces y protocolos del modelo OSI.

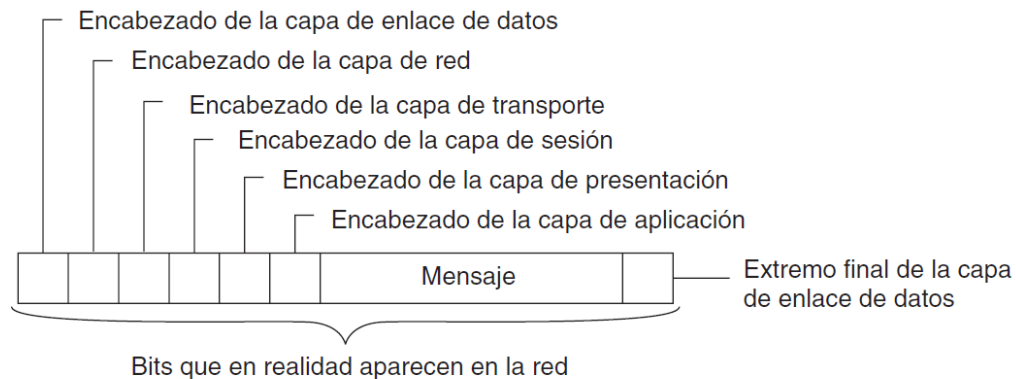


Figura 4-2. Mensaje típico de acuerdo a como aparece en la red.



Fundamentos de Comunicación

Protocolo en Capas: Modelo OSI → Protocolos de bajo nivel

- **Capa Física:** Maneja la estandarización eléctrica y mecánica así como la señalización de interfaces, de tal modo que cuando una máquina envía un bit 0, éste en realidad se recibe como un bit 0 y no como un bit 1.
- **Capa de Enlace Datos:** Agrupa los bits en **tramas** y ve que cada trama se reciba correctamente mediante un **checksum**.
- **Capa de Red:** Elige la mejor ruta para el mensaje enviado. El protocolo más utilizado es el **IP**.



Fundamentos de Comunicación

Protocolo en Capas: Modelo OSI → Capa de Transporte

- Divide el mensaje en piezas secuenciales.
- Ordena los mensajes que llegan para reconstruir el mensaje original.
- Protocolo de internet orientado a conexión es **TCP**.
- La combinación TCP-IP se utiliza ahora como un estándar predeterminado para comunicación en red
 - También existe orientado a no conexión llamado **UDP**.
- Existen otros protocolos de transporte como **RTP**, utilizado para transferencia de datos en tiempo real.



Fundamentos de Comunicación

Protocolo en Capas: Modelo OSI → Protocolos de alto nivel

- **Capa de Sesión:** Básicamente, es una versión mejorada de la capa de transporte. Proporciona herramientas de **sincronización** y proporciona control sobre el **dialogo**.
- **Capa de Presentación:** Otorga **significado** a las tramas de bits. Permite variabilidad en los sistemas que se comunican.
- **Capa de Aplicación:** Pensada para mantener aplicaciones estándar de red (correo, transferencia archivos, emulación de terminales). Se volvió el contenedor de toda aplicación y protocolo que no encaja en otra capa.



Fundamentos de Comunicación

Protocolo en Capas: Modelo OSI → Problemas

- Falta una clara diferencia entre aplicaciones, protocolos específicos de aplicaciones y protocolos de propósito general.
- **Protocolo de Transferencia de Archivos (FTP)**
 - Protocolo para transferencia entre un cliente y un servidor.
 - Programa FTP, aplicación para transferencia de archivos.
 - También se utiliza para implementar el FTP de internet.



Fundamentos de Comunicación

Existen protocolos de propósito general que son útiles para muchas aplicaciones, pero no pueden clasificarse en como protocolos de transporte.

Protocolos Middleware

- Reside en la capa de aplicación.
- Contiene protocolos de propósito general que garantiza sus propias capas.
- Soportan servicios de comunicación de alto nivel.
- Podría pertenecer a la capa de transporte, pero son igual de cercanos a la capa de aplicación.
 - Por ejemplo, broadcast sólo se puede implementar si se consideran los requerimientos de la aplicación.

Fundamentos de Comunicación

Protocolos Middleware

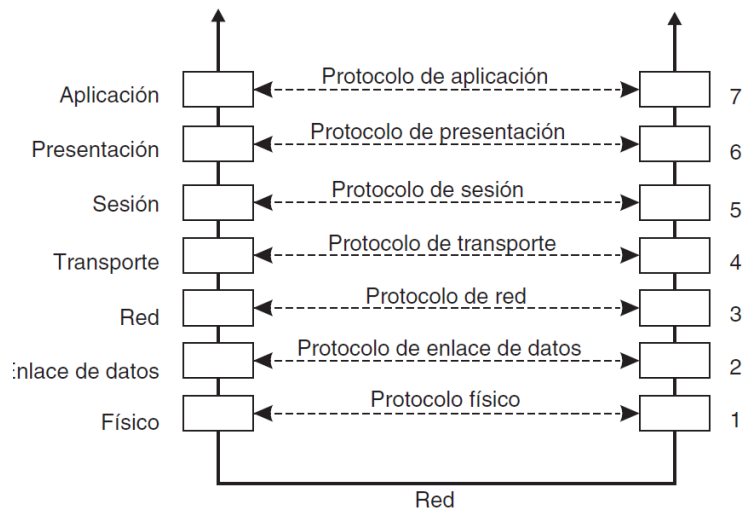


Figura 4-1. Capas, interfaces y protocolos del modelo OSI.

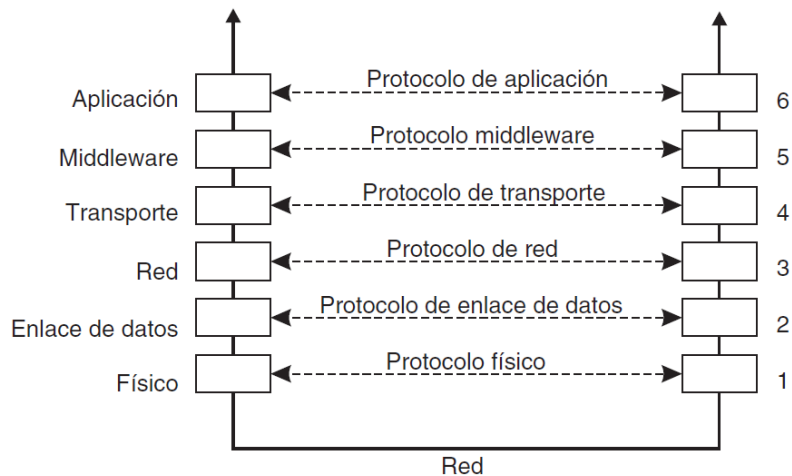


Figura 4-3. Modelo de referencia adaptado para comunicación en red.

Fundamentos de Comunicación

Tipos de Comunicación

- **Comunicación Persistente:** Middleware de comunicación almacena el mensaje y luego lo entrega. No existe necesidad de una conexión directa ni ejecución de aplicación destinataria.
- Los sistemas de correo electrónico usan comunicación persistente.

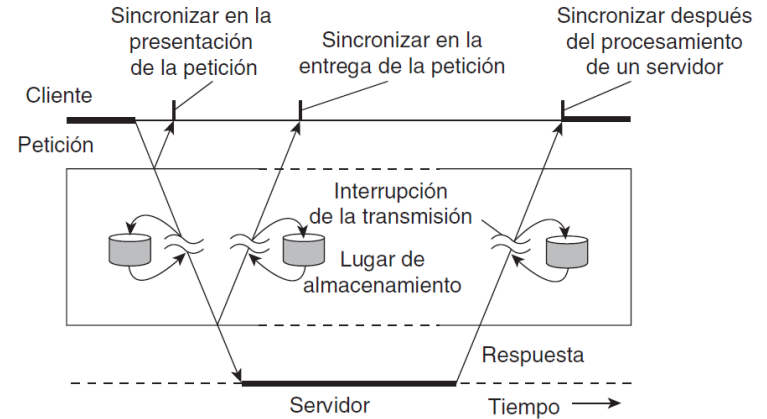


Figura 4-4. Perspectiva del middleware como un servicio intermediario (distribuido) al nivel de comunicación de aplicaciones.



Fundamentos de Comunicación

Tipos de Comunicación

- **Comunicación transitoria:** Almacena el mensaje sólo mientras las aplicaciones remitente y destinatario se ejecutan.
- **Comunicación asíncrona:** El remitente continua inmediatamente después de mandar el mensaje.
- **Comunicación sincrónica:** El remitente se bloquea hasta obtener una respuesta sobre su mensaje.



Llamadas a Procedimientos Remotos (RPC)

- Muchos sistemas distribuidos se han basado en el intercambio explícito de mensajes.
 - Los procedimientos *send* y *receive* no ocultan en absoluto la comunicación, lo cual es importante para lograr transparencia.
- Con las RPC se permite a la máquina A llamar a un procedimiento de la máquina B.
 - Quien programa no se preocupa de la comunicación, ya que es parte de la RPC misma.
- Ampliamente utilizadas en comunicación **cliente-servidor**.



Llamadas a Procedimientos Remotos (RPC)

- La idea tras las RPC es hacerla parecer como una **llamada local**.
 - Básicamente, la llamada del procedimiento local no debe advertir que la ejecución del procedimiento se realiza en una máquina distinta.
- Las RPC aparecen como **interfaces**.
 - Enmascara la llamada y permite cambiar las funcionalidades sin afectar la forma en que se llama.
 - En un modelo C-S, el servidor proporciona a los clientes un listado de módulos que puede utilizar con los parámetros necesarios.



Llamadas a Procedimientos Remotos (RPC)

- La RPC busca **transparencia**.
 - Al hacer la llamada del procedimiento remoto lo más parecido a una llamada local.
 - Ofrece transparencia de localización y acceso, ocultando la localización física de los métodos.
- La llamada a un RPC es más **propensa a fallos**, por su necesidad de conexión en red.
 - Problemática para determinar la razón del fallo.



Llamadas a Procedimientos Remotos (RPC)

- Existen diferentes métodos para garantizar la entrega de mensajes mediante RPC.
- **Tal vez:** El RPC puede ejecutarse una vez o ninguna. El cliente puede o no recibir respuesta. El cliente envía una petición y queda en espera durante un tiempo t , si la respuesta no llega en dicho tiempo, continúa su ejecución y, por tanto, el cliente no tiene retroalimentación en caso de fallo.
- **Al menos una vez:** El RPC se ejecuta una o más veces. El cliente puede recibir una o más respuestas. El cliente espera un tiempo t , si no tiene respuesta, reenvía la petición. El servidor no descarta duplicados, por lo que el cliente puede recibir varias respuestas. Sólo aplicable para operaciones **idempotentes**.
- **Como máximo una vez:** El RPC se ejecuta una vez o no se ejecuta. El cliente recibe respuesta o indicación de que no fue posible ejecutar el RPC. Cliente espera tiempo t y repite petición. Servidor filtra peticiones duplicadas y sólo se ejecuta una vez.

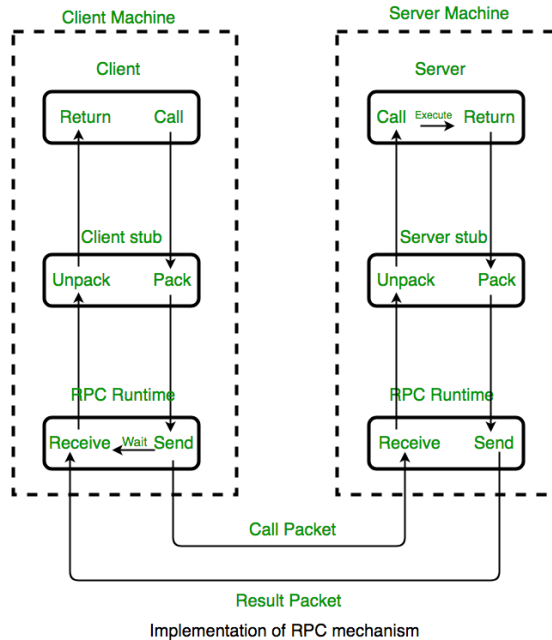


Llamadas a Procedimientos Remotos (RPC)

Semántica	Cliente	Servidor	Reintentar solicitud	Filtro duplicados	Retransmisión de resultados
Tal vez	Espera tiempo t y luego descarta petición	Ejecuta petición y envía respuesta	No	No	No
Al menos una vez	Espera tiempo t y luego repite petición	Ejecuta todas las solicitudes y da respuesta a cada una	Si	No	Ejecuta nuevamente el RPC
Como máximo una vez	Espera tiempo t y luego repite petición	Ejecuta solicitud y descarta duplicados	Si	Si	Retransmite respuesta (servidor con memoria)



Llamadas a Procedimientos Remotos (RPC)





Comunicación Orientada a Mensajes

- RPC requiere que ambas máquinas estén activas para generar la comunicación.
- RPC en general es síncrono.
- Alternativa, comunicación orientada a mensajes.
- Utiliza primitivas de sockets para generar la comunicación.
- Los sockets son limitantes e insuficientes.
 - Diseñados para comunicación TPC/IP pero no para redes de alta velocidad.
 - Malo para HPC.



Comunicación Orientada a Mensajes

- **MPI**, interfaz para el paso de mensajes que establece un estándar para el paso de mensajes.
- Asume que la comunicación ocurre dentro de un grupo conocido de procesos.
- Define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes, diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.
- Su principal característica es que no precisa de memoria compartida, por lo que es muy importante en la programación de sistemas distribuidos.



Comunicación Orientada a Mensajes

- Comunicación persistente orientada a mensajes se establece en **Middleware Orientado a Mensajes (MOM)**.
- Se implementa en sistemas de colas de mensajes.
 - No requiere que el remitente y/o el destinatario estén activos al momento de enviar el mensaje.
- Dirigido para transferencia de mensajes que duran minutos en lugar de segundos o milisegundos.

Comunicación Orientada a Mensajes

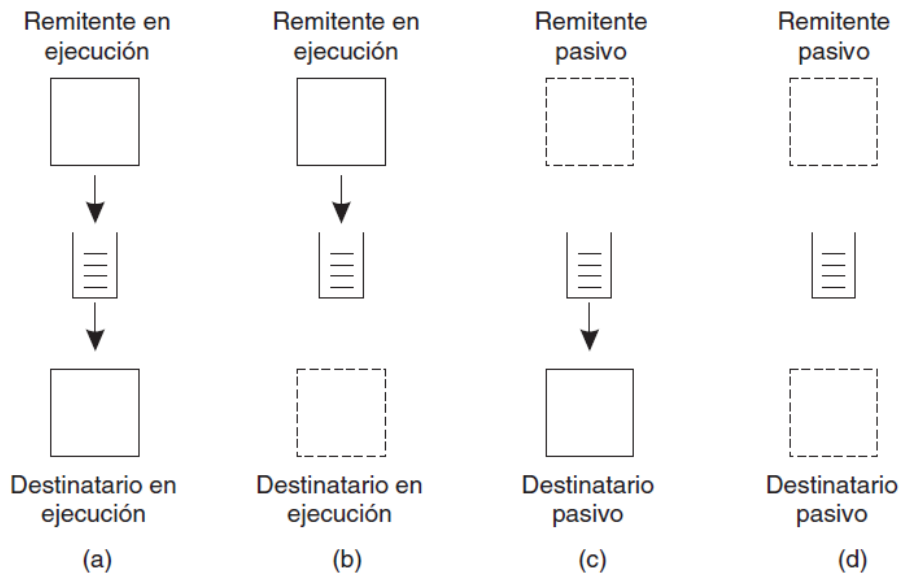


Figura 4-17. Cuatro combinaciones de comunicaciones muy poco acopladas mediante el uso de colas.



Comunicación Orientada a Flujo (Streaming)

Utilizada para comunicación que depende del tiempo y la sincronización para lograr calidad en el mensaje.

Soporte para Medios Continuos

- Soporte para información dependiente del tiempo.
- Para los medios es importante la forma en que se representa los datos.
- Relaciones temporales para interpretar correctamente lo que significan los datos.
- Por contraste, los **medios discretos**, no dependen de las relaciones temporales para interpretar de manera correcta los datos.



Comunicación Orientada a Flujo (Streaming)

Flujo de Datos

- El flujo de datos es una secuencia de unidades de datos.
- **Sincronización** es crucial para los flujos de continuos de datos.
- **Transmisión asíncrona**: Elementos transmitidos uno tras otro, pero no hay restricciones de sincronización en cuanto al tiempo de la transmisión.
- **Transmisión síncrona**: Existe un retraso máximo fin a fin para cada unidad del flujo de datos.
- **Transmisión isócrona**: Las unidades deben ser transmitidas a tiempo. Existe un retraso mínimo y uno máximo fin a fin, conocido como **inestabilidad limitada**.



Comunicación Orientada a Flujo (Streaming)

Flujo de Datos

- **Flujo Simple:** Consta únicamente de una secuencia de datos.
- **Flujo Complejo:** Consta de varios flujos simples relacionados denominados **subflujos**.
- Transmisión de películas, música, vídeos, entre otros, son ejemplos de flujo complejo.
- El flujo puede ser en vivo o ser un flujo de datos almacenados.
- Flujo en vivo brinda menos oportunidades de ajuste, por lo que nos centramos en flujo de datos almacenados.

Comunicación Orientada a Flujo (Streaming)

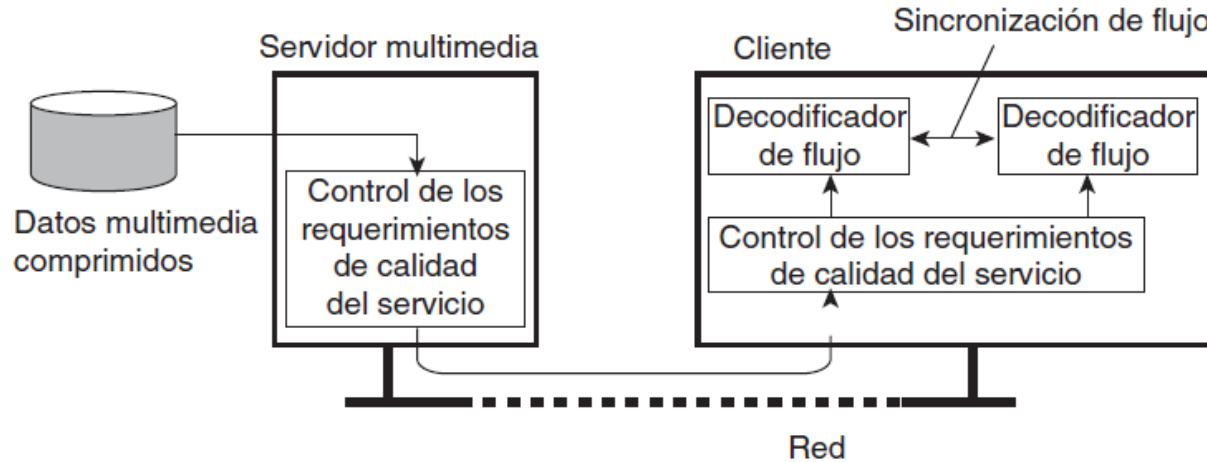


Figura 4-26. Arquitectura general para pasar a través de una red un flujo de datos multimedia almacenados.



Comunicación Orientada a Flujo (Streaming)

Calidad de Servicio

- Habla sobre los requerimientos de sincronización.
- Para asegurar sincronización, existen técnicas como **reenvío expedito** y **reenvío garantizado**.
 - Definen prioridades y las redes que se utilizarán.
 - Uso de buffers.

Comunicación Orientada a Flujo (Streaming)

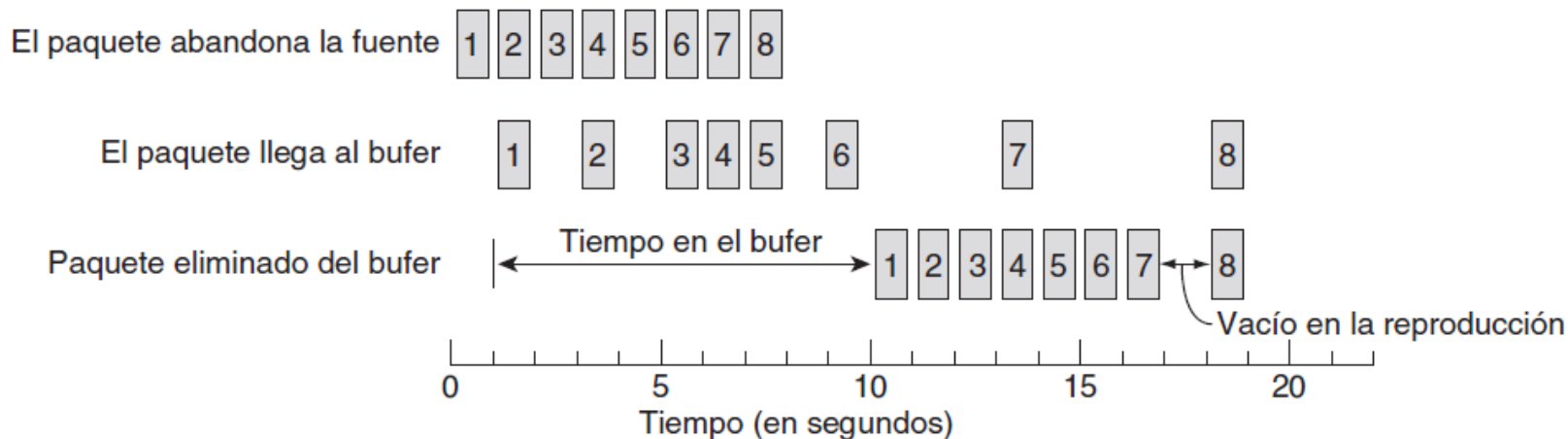


Figura 4-27. Uso de un bufer para reducir la inestabilidad.

Comunicación Orientada a Flujo (Streaming)

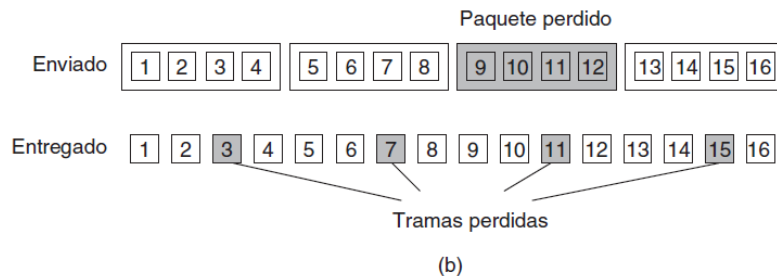
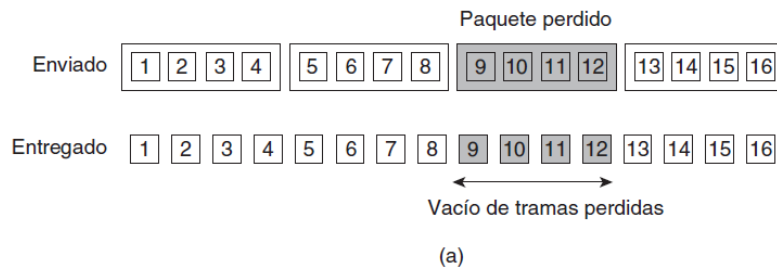


Figura 4-28. Efecto de un paquete perdido en (a) una transmisión no interpolada, y en (b) una transmisión interpolada.

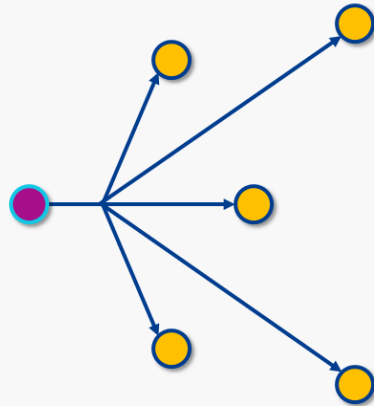


Comunicación por Multitransmisión

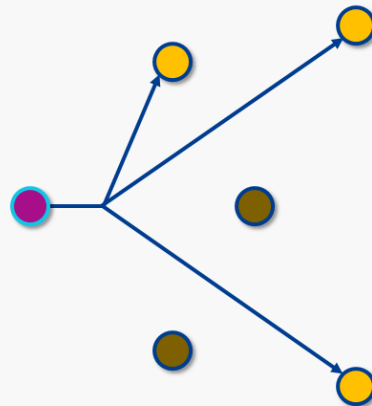
- Envío de información en múltiples redes a múltiples destinos simultáneamente.
- Para efectuar este tipo de transmisión se debe establecer un grupo de multitransmisión.
- El grupo tiene asociada una dirección en la red y el emisor envía el mensaje a uno o varios grupos.
 - Cada proceso puede pertenecer a más de un grupo.
- No confundir con **broadcast**.

Comunicación por Multitransmisión

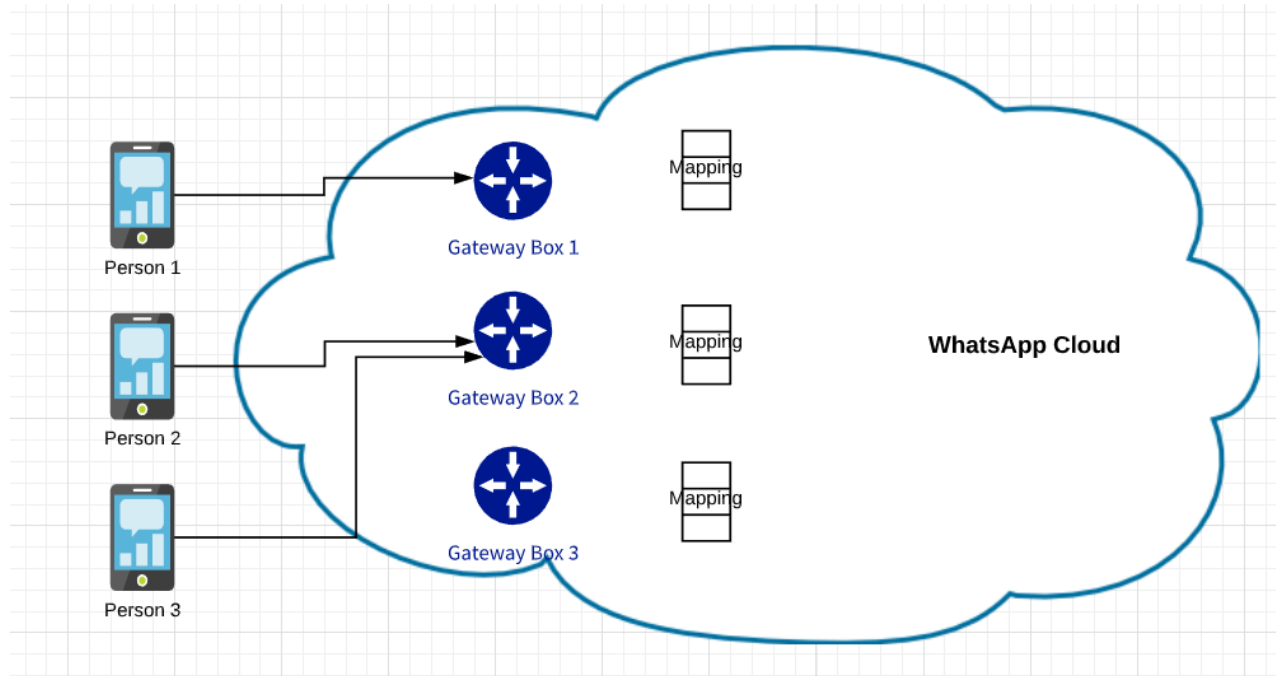
Broadcast



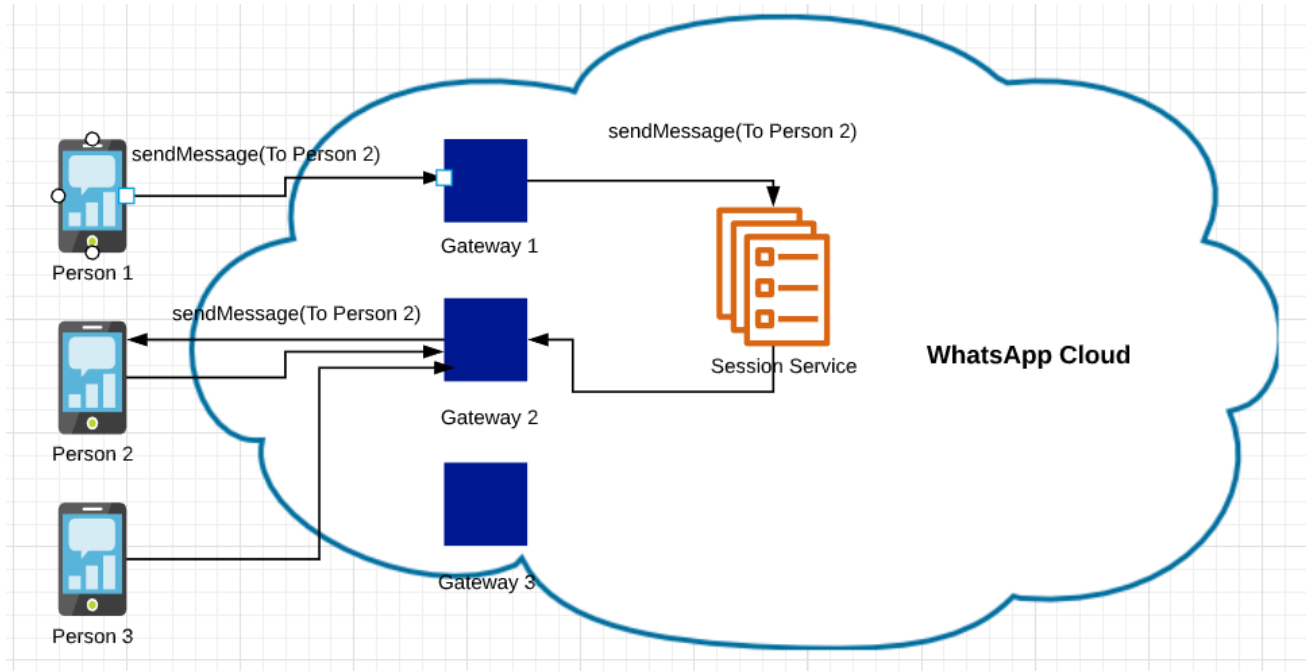
Multicast



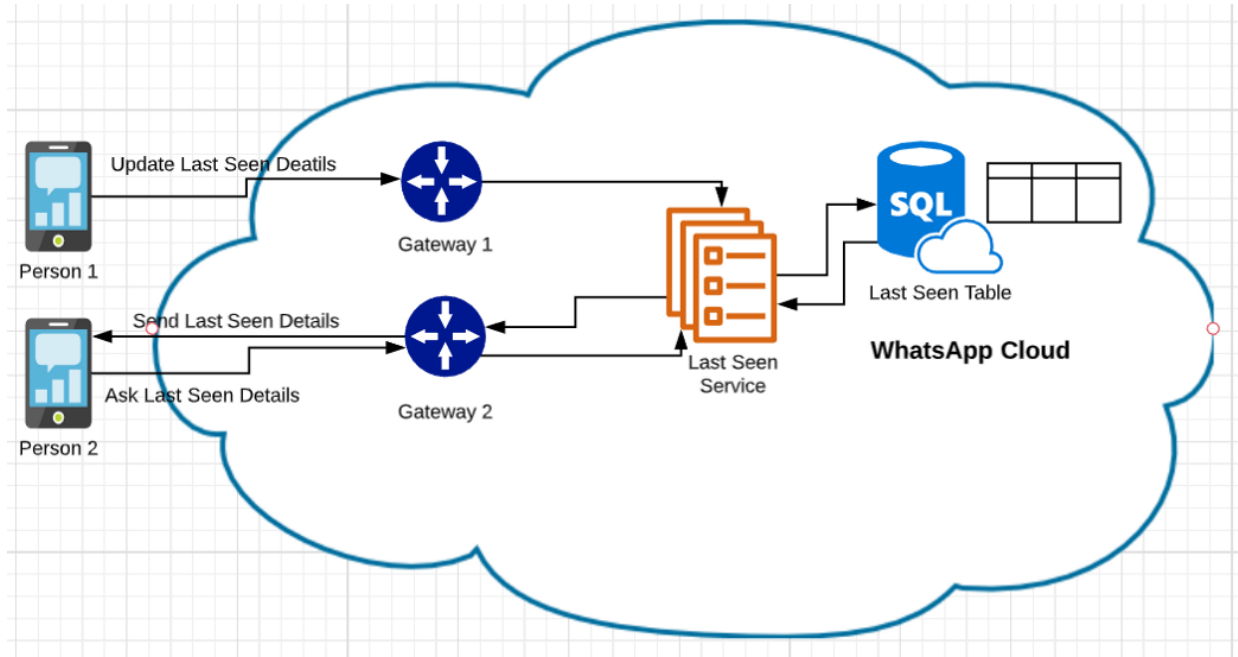
Comunicación en Whatsapp



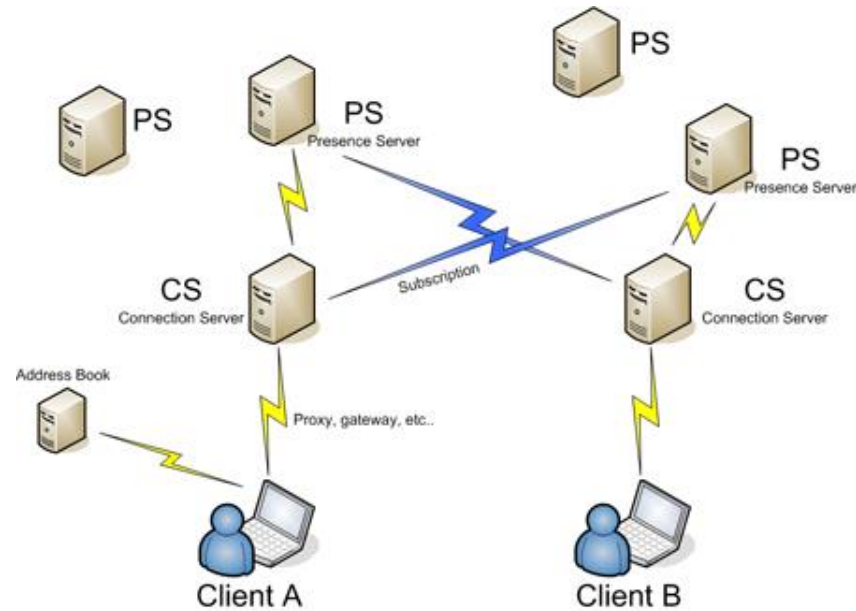
Comunicación en Whatsapp



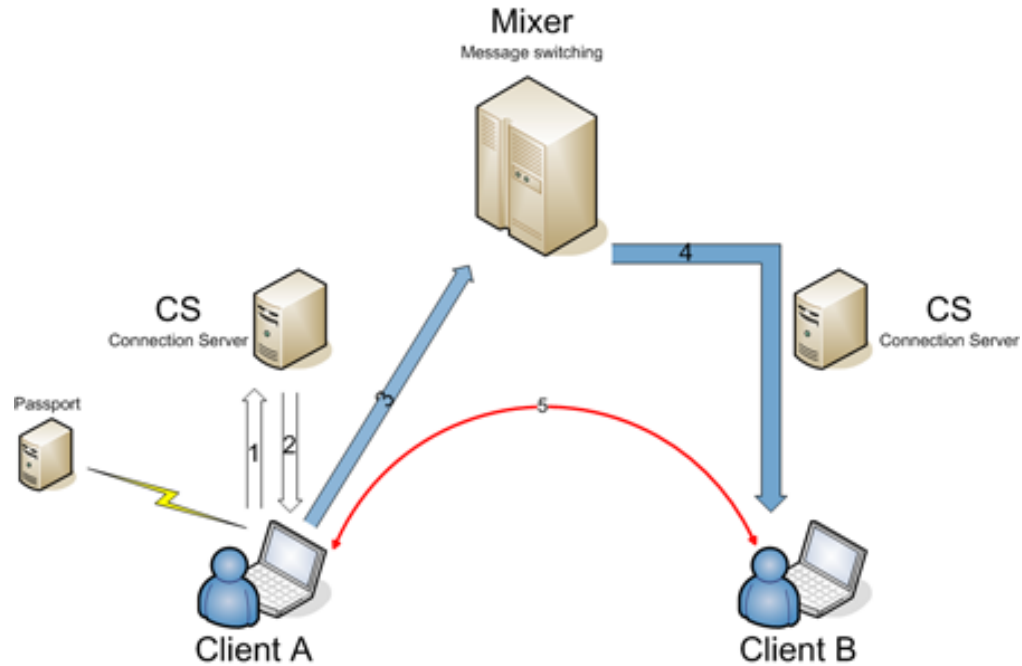
Comunicación en Whatsapp



Comunicación en Windows Live Messenger



Comunicación en Windows Live Messenger





DEPARTAMENTO DE
**INGENIERÍA
INFORMÁTICA**
UNIVERSIDAD DE SANTIAGO DE CHILE

Departamento de Ingeniería Informática
Universidad de Santiago de Chile

¿CONSULTAS?