UNIVERSIDAD DIEGO PORTALES · ESCUELA DE INFORMÁTICA Y TELECOMUNICACIONES

Redes de Datos
Material de Estudio Integral

CIT2114 — Guía completa para preparación de evaluaciones

Modelos OSI / TCP-IP | Capa Física & Enlace | Direccionamiento IPv4 / IPv6 | VLSM | Enrutamiento | Capa de Transporte

Foco especial: Solemne 2 — ejercicios resueltos paso a paso

Material orientado a estudiantes · Elaborado a partir de las guías oficiales del curso

Cómo usar esta guía

Está organizada siguiendo el orden del semestre, de los fundamentos a lo más complejo. Cada sección combina teoría esencial, fórmulas clave y ejercicios. La **Sección 8 (Solemne 2)** incluye **todos los ejercicios resueltos y verificados**, ideal para practicar antes de la evaluación. Las cajas de color indican: azul = concepto clave, verde = tip/respuesta correcta, naranjo = error común que debes evitar.

Leyenda de las cajas que verás en esta guía:

Concepto clave (azul)

Resalta definiciones e ideas centrales que debes dominar sí o sí.

Tip / Respuesta (verde)

Consejos prácticos, atajos de cálculo y las respuestas correctas de los ejercicios.

Atención (naranjo)

Errores frecuentes y confusiones típicas que debes evitar en la evaluación.

Checklist crítico (rojo)

Reglas que, si las olvidas, invalidan tu resultado.

Índice de contenidos

1. Fundamentos de Redes y Modelos de Referencia

Una **red de datos** es un conjunto de dispositivos interconectados que intercambian información, generalmente de forma bidireccional y en tiempo real. Se compone de **dispositivos** (routers, switches, módems, hosts) y **medios de transmisión** (cobre, fibra óptica o el aire).

Conceptos transversales

- **Direccionamiento:** mecanismo para identificar un destino específico cuando hay múltiples receptores.
- **Control de flujo:** evita que un emisor rápido sature a un receptor lento, mediante retroalimentación o velocidades acordadas.
- **Control de errores:** los circuitos físicos no son perfectos; ambos extremos acuerdan un código de detección.
- **Mantenimiento de secuencia:** numeración que permite reordenar los mensajes si llegan fuera de orden.
- **Abstracción por capas:** cada capa cumple una función bien definida para minimizar el flujo de información entre interfaces.

El Modelo OSI (7 capas)

Marco de referencia de 7 capas, creado *antes* de que existieran los protocolos. Cada capa tiene su propia unidad de datos o **PDU** (Protocol Data Unit).

#	Capa	Función principal	PDU
7	**Aplicación**	Servicios de red directos al usuario (HTTP, DNS, FTP...)	Datos
6	**Presentación**	Traduce, representa, comprime y **cifra** los datos	Datos
5	**Sesión**	Establece, gestiona y finaliza diálogos (Simplex/Half/Full-Duplex)	Datos
4	**Transporte**	Comunicación extremo a extremo, fiabilidad (TCP/UDP)	**Segmento**
3	**Red**	Direccionamiento lógico (IP) y determinación de ruta	**Paquete**
2	**Enlace**	Direccionamiento físico (MAC), acceso al medio	**Trama**
1	**Física**	Transmisión de bits sobre el medio físico	**Bits**

> **Regla mnemotécnica (capa 7 → 1)**
>
> **A**plicación · **P**resentación · **S**esión · **T**ransporte · **R**ed · **E**nlace · **F**ísica. Y para las PDU de las capas 4-3-2-1: **S**egmento → **P**aquete → **T**rama → **B**its.

Diagrama: capas y su PDU

A medida que los datos **bajan** por el emisor, cada capa agrega su encabezado (encapsulamiento) y la unidad cambia de nombre. En el receptor el proceso es inverso (desencapsulamiento).

Capa	Nombre	PDU (nombre del dato)	Ejemplo de dato
7	Aplicación	**Datos** (mensaje)	Petición HTTP, correo
6	Presentación		Datos cifrados/comprimidos
5	Sesión		Diálogo de la sesión
4	Transporte	**Segmento** (TCP) / Datagrama (UDP)	Datos + puertos
3	Red	**Paquete**	Segmento + IP origen/destino
2	Enlace	**Trama**	Paquete + MAC origen/destino + FCS
1	Física	**Bits**	Señal eléctrica/óptica/radio

Flujo de encapsulamiento (emisor → medio)

Datos → [+cabecera de transporte] **Segmento** → [+cabecera IP] **Paquete** → [+cabecera y cola de enlace] **Trama** → **Bits** sobre el medio.

Modelo TCP/IP (4 capas)

A diferencia de OSI, en TCP/IP los **protocolos se desarrollaron primero** y el modelo fue una descripción posterior de ellos. Sus 4 capas son: Aplicación, Transporte, Internet y Acceso a la Red.

Diferencia clave OSI vs TCP/IP

OSI es un modelo **teórico/de referencia** (7 capas, creado antes que los protocolos). TCP/IP es un modelo **práctico** (4 capas, los protocolos vinieron primero). OSI separa Sesión y Presentación; TCP/IP las agrupa dentro de Aplicación.

2. Capa Física: Medios y Transmisión

Medios de transmisión guiados e inalámbricos

- **Cable UTP (par trenzado):** el más usado en LAN. Sus pares se trenzan para reducir interferencia y ruido EMI/RFI. Normas de cableado: **T568A** y **T568B** (ANSI/TIA-568).
- **Cable coaxial:** menos susceptible al ruido que el UTP gracias a su malla conductora externa que actúa como pantalla.
- **Fibra óptica:** preferida para interconectar edificios: mayores distancias, baja susceptibilidad a interferencias y alto ancho de banda.
- **Banda ISM:** banda de frecuencia para usos Industriales, Científicos y Médicos (inalámbrico).

Variantes del cable de par trenzado (blindaje)

No todos los cables de par trenzado son iguales: se diferencian por el **blindaje** que llevan contra interferencias electromagnéticas. La nomenclatura ISO/IEC 11801 indica el blindaje general/del par así: `(blindaje exterior)/(blindaje por par)TP`.

Tipo	Nombre	Blindaje	Uso típico
UTP	Unshielded Twisted Pair	Sin blindaje, solo el trenzado	LAN de oficina/hogar (lo más común y económico)
FTP	Foiled Twisted Pair (F/UTP)	Una **lámina de aluminio (foil)** general que envuelve todos los pares	Entornos con interferencia moderada
STP	Shielded Twisted Pair	Blindaje (malla y/o foil) que protege los pares	Entornos industriales con alta EMI/RFI
S/FTP	Screened/Foiled TP	Malla general **+** foil individual por par (doble blindaje)	Máxima protección (datacenter, 10GBASE-T)

> **¡No confundas las dos "STP"!**
> En cableado, **STP = Shielded Twisted Pair** (un tipo de cable). En conmutación de capa 2, **STP = Spanning Tree Protocol** (evita bucles). Son conceptos totalmente distintos que comparten sigla.

Medios inalámbricos: estándar IEEE 802.11 (Wi-Fi)

El estándar **IEEE 802.11** define las redes WLAN (Wi-Fi). Su "medio de transmisión" no es un cable sino el **aire (espacio libre)**, por donde viajan ondas de radio en bandas **ISM** no licenciadas. Características clave:

- **Bandas de frecuencia:** principalmente **2,4 GHz** (mayor alcance, más interferencia y saturación) y **5 GHz** (más velocidad y canales, menor alcance). Wi-Fi 6E/7 incorpora la banda de **6 GHz**.
- **Acceso al medio CSMA/CA:** a diferencia de Ethernet (CSMA/CD), el Wi-Fi usa **Collision Avoidance**: como el nodo no puede "escuchar mientras transmite" en un medio compartido por radio, **evita** la colisión esperando turnos y confirmando con ACK, en lugar de detectarla.
- **Medio compartido y half-duplex:** todos los dispositivos comparten el canal; solo uno transmite a la vez por canal.
- **Principales enmiendas:** 802.11n (Wi-Fi 4), 802.11ac (Wi-Fi 5), 802.11ax (Wi-Fi 6) — cada una aumenta velocidad y eficiencia (MIMO, OFDMA).

Tipos de cable UTP

Tipo	Normas en los extremos	Se usa para conectar
Directo	Misma norma en ambos extremos (T568B – T568B)	Dispositivos **diferentes** (PC ↔ Switch, Switch ↔ Router)
Cruzado	Normas distintas (T568A – T568B)	Dispositivos **iguales** (PC ↔ PC, Switch ↔ Switch)

Cálculos críticos

Capacidad de Shannon: $C = B \cdot \log_2(1 + SNR)$

Determina la capacidad máxima teórica de un canal. Significado de cada parte:

- **C** = capacidad máxima del canal, en **bits por segundo (bps)**. Es el resultado que se busca.
- **B** = ancho de banda del canal, en **hertz (Hz)**. Mientras mayor sea, más información cabe.
- **SNR** = relación señal/ruido en **valor absoluto** (veces, NO en dB). Indica cuántas veces más fuerte es la señal que el ruido.

- **$\log_2(...)$** = logaritmo en base 2; aparece porque cada bit duplica la cantidad de estados que la señal puede distinguir.

> **Error frecuente con Shannon**
>
> Si la SNR viene en **decibelios (dB)**, primero conviértela a valor absoluto antes de usar la fórmula: `SNR = 10^(dB/10)` . Ejemplo: 30 dB → $10^3 = 1000$.

Apertura numérica (fibra): $NA = \sqrt{(n_1^2 - n_2^2)}$

Mide qué tanta luz puede captar y guiar la fibra. Significado de cada parte:

- **NA** = apertura numérica (valor **sin unidad**): a mayor NA, mayor el ángulo de luz que la fibra acepta.
- **n_1** = índice de refracción del **núcleo** (la parte central por donde viaja la luz).
- **n_2** = índice de refracción del **revestimiento/cladding** (la capa que envuelve el núcleo).
- Siempre **$n_1 > n_2$**, condición necesaria para que la luz quede atrapada por reflexión interna total.

3. Capa de Enlace y Protocolos de Acceso

Dispositivos y dominios

Dispositivo	Capa	¿Qué segmenta?
Hub	1 (Física)	No segmenta nada: amplifica la señal. Un solo dominio de colisión y de broadcast.
Switch	2 (Enlace)	Segmenta **dominios de colisión** (uno por puerto). Filtra por MAC.
Router	3 (Red)	Segmenta **dominios de broadcast** y conecta redes distintas.

Bridge (puente) vs Switch

El **bridge** es el antecesor del switch. Ambos trabajan en **capa 2**, reenvían tramas según la dirección **MAC** y segmentan dominios de colisión, pero hay diferencias clave:

Aspecto	Bridge (puente)	Switch
Procesamiento	Por **software** (más lento)	Por **hardware/ASIC** (a velocidad de cable)
Nº de puertos	Pocos (2–4)	Muchos (24, 48 o más)
Dominios de colisión	Uno por puerto (pocos)	Uno por puerto (muchos)
Uso actual	Obsoleto/legado	Estándar en toda LAN moderna

> **Idea para recordar**
>
> Un **switch es esencialmente un "bridge multipuerto"** implementado en hardware. Hace lo mismo que el bridge (aprender MAC, filtrar y reenviar), pero con muchos más puertos y mucha más velocidad.

Direcciones MAC y tablas

- **MAC:** dirección física, grabada en la NIC, de **48 bits (6 octetos)**. Se divide en dos mitades de 24 bits:
  - **Primeros 3 octetos = OUI** (Organizationally Unique Identifier): identifican al **fabricante**, asignados por el IEEE (ej. `00:1A:2B`).
  - **Últimos 3 octetos = identificador del dispositivo** (NIC Specific): los asigna el propio fabricante para que **cada tarjeta sea única** en el mundo. Juntos garantizan una MAC irrepetible.
- **Tabla ARP:** mapea direcciones **IP → MAC**.
- **Tabla MAC (CAM):** relaciona **puertos del switch → direcciones MAC**.

> **Encapsulamiento en cada salto (¡muy preguntado!)**
> En cada salto de router, las direcciones **MAC de origen y destino cambian**, pero las **direcciones IP de origen y destino se mantienen** de extremo a extremo.

Spanning Tree Protocol (STP)

- **Función:** elimina bucles (loops) en topologías redundantes creando una estructura lógica en árbol.
- **Root Bridge:** se elige el switch con el **Bridge ID más bajo** (prioridad y, si empata, la MAC más baja).
- **Roles de puerto:** Root Port (RP), Designated Port (DP) y Non-Designated/Blocked (NDP).

Mecanismos de control de acceso al medio

- **CSMA/CD:** en Ethernet, detecta colisiones; el emisor "escucha" el medio antes y durante la transmisión.
- **Back-off exponencial:** tras una colisión, cada nodo espera un tiempo aleatorio creciente para reducir la probabilidad de una nueva colisión.

VLANs y diseño jerárquico

- **VLAN:** agrupa dispositivos de forma lógica; mejora la seguridad y reduce el tamaño de los dominios de broadcast.
- **VLAN estática vs dinámica:** en la **estática** (la más común) el administrador asigna manualmente cada puerto del switch a una VLAN. En la **dinámica**, el dispositivo se asigna a su VLAN **automáticamente según su identidad** (su dirección MAC o las credenciales de usuario), sin importar a qué puerto se conecte; esto se gestiona con autenticación **802.1X + servidor RADIUS** (históricamente con VMPS en Cisco). Útil en entornos con usuarios móviles.
- **Modelo jerárquico de 3 capas: Núcleo/Core** (conmutación de alta velocidad) → **Distribución** (agregación y políticas) → **Acceso** (conexión de usuarios finales).
- **ACL:** reglas para permitir o denegar tráfico según IP y protocolo.

4. Direccionamiento IPv4 e IPv6

Direccionamiento por clases (classful) y por qué ya no se usa

Originalmente IPv4 dividía todo el espacio de direcciones en **clases fijas** según los primeros bits del primer octeto. Cada clase traía una **máscara fija** obligatoria:

Clase	1er octeto	Máscara fija	Uso
A	1 – 126	/8	Redes muy grandes (~16,7 M hosts)
B	128 – 191	/16	Redes medianas (~65 mil hosts)
C	192 – 223	/24	Redes pequeñas (254 hosts)
D	224 – 239	—	Reservada para **multicast**
E	240 – 255	—	Reservada (experimental)

Hoy el modelo classful está obsoleto — ¿por qué?

Las clases fijas **desperdiciaban enormes cantidades de direcciones**: una empresa con 300 hosts necesitaba más de un /24, pero un /16 (clase B) le daba 65.534 (miles desaprovechados). Esto, sumado al **agotamiento de IPv4** y al crecimiento de las tablas de enrutamiento, llevó a adoptar **CIDR** (Classless Inter-Domain Routing, RFC 1518/1519, 1993): se usa una **máscara de longitud variable** (el "/n") en lugar de la clase, permitiendo asignar exactamente lo necesario y agregar rutas (sumarización). Por eso VLSM y CIDR son el estándar actual y el classful solo se estudia como base histórica.

Nota: el rango `127.0.0.0/8` está reservado para loopback, por eso la clase A llega hasta 126.

Clases y rangos privados IPv4

Clase	Rango privado completo	Máscara por defecto
A	10.0.0.0 – 10.255.255.255	/8 (255.0.0.0)
B	172.16.0.0 – 172.31.255.255	/12 (255.240.0.0)
C	192.168.0.0 – 192.168.255.255	/16 (255.255.0.0)

Cuidado con las fronteras

Cualquier dirección fuera de estos rangos es **pública**. Ejemplos clásicos de examen: `172.15.5.1` y `192.167.1.1` son **públicas** (quedan justo fuera de las fronteras privadas de B y C). También son especiales: `127.0.0.0/8` (loopback) y `169.254.0.0/16` (APIPA / link-local).

Tipos de direcciones IPv4

Tipo	¿A quién llega?	Ejemplo / rango
Unicast	A **un único** host destino (comunicación 1 a 1)	192.168.1.10
Broadcast	A **todos** los hosts de la red (1 a todos)	Dirigido: x.x.x.255 · Limitado: 255.255.255.255
Multicast	A un **grupo** suscrito de hosts (1 a muchos)	224.0.0.0 – 239.255.255.255 (clase D)
Loopback	Al **propio host** (pruebas de la pila TCP/IP)	127.0.0.0/8 (típico 127.0.0.1)
Link-local (APIPA)	Autoasignada cuando no hay DHCP; solo válida en el enlace local	169.254.0.0/16

Tipos de direcciones IPv6 y diferencias con IPv4

IPv6 (128 bits) **reorganiza** los tipos de direcciones. La diferencia más importante: **IPv6 elimina el broadcast** y lo reemplaza por multicast, e incorpora el anycast.

Tipo	¿A quién llega?	Notas
Unicast	A una **única** interfaz (1 a 1)	Global (2000::/3), Link-local (fe80::/10), Loopback (::1), Unique-local (fc00::/7)
Multicast	A un **grupo** de interfaces suscritas (1 a muchos)	ff00::/8. Reemplaza al broadcast. Ej: `ff02::1` = todos los nodos del enlace
Anycast	A la interfaz **más cercana** de un grupo (1 al "mejor de varios")	Útil para servidores replicados (ej. DNS, CDN)

> **¿Por qué IPv6 NO necesita broadcast?**
> El broadcast de IPv4 es **ineficiente**: interrumpe a **todos** los dispositivos del segmento aunque el mensaje no les interese, consumiendo CPU y ancho de banda. IPv6 lo sustituye por **multicast dirigido a grupos específicos**: por ejemplo, en vez de un ARP por broadcast, IPv6 usa **NDP (Neighbor Discovery Protocol)** con multicast de nodo solicitado, de modo que **solo el host relevante** recibe y procesa el mensaje. Resultado: menos tráfico inútil y mejor rendimiento.

Simplificación de direcciones IPv6

IPv6 usa **128 bits** en 8 bloques hexadecimales (hextetos) de 4 dígitos. Dos reglas para compactar:

1. **Omitir ceros a la izquierda** de cada hexteto: `0DB8 → DB8`, `0000 → 0`, `001F → 1F`.
2. **Doble dos puntos** `::`: reemplaza **una sola** secuencia consecutiva de hextetos todos en cero. Solo se puede usar **una vez** por dirección (si hay varias secuencias, se comprime la más larga).

Ejemplo resuelto:

`2001:0DB8:0000:4A00:0000:0000:0000:001F`

1) Quitar ceros iniciales → `2001:db8:0:4a00:0:0:0:1f`

2) Comprimir la secuencia más larga de ceros (los tres hextetos finales) con `::`

Resultado: **2001:db8:0:4a00::1f**

Nota: el cero suelto entre `db8` y `4a00` queda como `0`, porque `::` ya se usó en la secuencia más larga (no puede repetirse).

Configuración automática EUI-64

Permite que un host genere los **64 bits del identificador de interfaz** a partir de su MAC de 48 bits.

1. Dividir la MAC en dos mitades de 24 bits.
2. Insertar **FFFE** en el medio.
3. **Invertir el 7º bit** del primer byte (bit U/L). En la práctica: si el 2º dígito hex es par se le suma 2 (00→02); el resultado es el bit Universal/Local invertido.

Ejemplo: Prefijo `2001:DB8:FACE:1::/64`, MAC `00:E0:4C:11:22:33`

Insertar FFFE → `00E0:4CFF:FE11:2233`

Invertir 7º bit del primer byte: `00` (00000000) → `02` (00000010)

Dirección final: **2001:DB8:FACE:1:2E0:4CFF:FE11:2233**

5. Subneteo: FLSM y VLSM

Fórmulas base

```
Hosts útiles por subred = 2^(bits de host) − 2   |   Nº de subredes = 2^(bits robados)   |   Incremento = 256 − (valor del octeto de la máscara)
```

Significado de cada parte:

- **bits de host** = bits que quedan para numerar hosts (los ceros de la máscara) = 32 − prefijo.
- **− 2** = se descartan 2 direcciones no asignables: la de **red** (primera) y la de **broadcast** (última).
- **bits robados** = bits que se le quitan a la porción de host para crear subredes nuevas.
- **octeto de la máscara** = valor decimal del octeto donde "cae" el prefijo; **256 − ese valor** da el **incremento** (el salto entre una subred y la siguiente).

FLSM — Subneteo tradicional (bloques iguales)

Problema: red 192.168.1.0/24, hallar la subred Nº 5 con máscara /28.

Incremento: /28 toma 4 bits del 4º octeto → 2^(8−4) = **16**. Las subredes saltan de 16 en 16.

Subred 0 = .0 · Subred 1 = .16 · ... · **Subred 5 = .80**

Parámetro	Valor
Dirección de red	192.168.1.80
Primer host útil	192.168.1.81
Último host útil	192.168.1.94
Broadcast	192.168.1.95

VLSM — Máscara de longitud variable

★ Regla de oro de VLSM

Ordena SIEMPRE los requerimientos de hosts de **MAYOR a MENOR** antes de asignar. Si no lo haces, los rangos se solapan y el direccionamiento queda inválido. Para cada subred, busca el menor **x** tal que **2^x − 2 ≥ hosts requeridos**; el prefijo será **/(32 − x)**.

Método paso a paso

1. Ordena las subredes de mayor a menor cantidad de hosts.
2. Para la primera, calcula bits de host (x) y máscara; asígnala desde la dirección base.
3. El **broadcast + 1** es la dirección de red de la siguiente subred.
4. Repite hasta cubrir todas las subredes, sin dejar espacios vacíos.

Ejemplo resuelto: Red base 10.10.10.0/24

LAN	Hosts	Red	Máscara	1er host	Último host	Broadcast
Contabilidad	110	10.10.10.0	/25 (255.255.255.128)	.1	.126	.127
Ventas	50	10.10.10.128	/26 (255.255.255.192)	.129	.190	.191
TI	12	10.10.10.192	/28 (255.255.255.240)	.193	.206	.207
Enlace WAN	2	10.10.10.208	/30 (255.255.255.252)	.209	.210	.211

Observa cómo el broadcast de cada subred + 1 da la red de la siguiente. El enlace WAN punto a punto siempre usa /30 (2 hosts útiles exactos).

6. Capa de Red: Enrutamiento

¿Cómo decide un router?

Un router usa **exclusivamente su Tabla de Enrutamiento** para determinar el mejor camino hacia una red de destino. **No** usa para ello la tabla ARP (IP/MAC), la tabla MAC (capa 2) ni DNS.

Enrutamiento estático vs dinámico

Ventajas del **estático**: mayor **seguridad** (no anuncia ni propaga rutas, evitando que nodos no autorizados vean la topología), **menor carga de CPU** y **ahorro de ancho de banda** (no envía actualizaciones periódicas). Desventaja: no escala bien y debe configurarse manualmente.

Sintaxis de ruta estática (Cisco IOS)

```
! Estructura general
ip route [Red_Destino] [Máscara] [IP_Siguiente_Salto]

! Ejemplo: alcanzar la LAN de Servidores 10.20.0.0/25 vía 172.16.10.2
Alpha(config)# ip route 10.20.0.0 255.255.255.128 172.16.10.2
```

Tip de conectividad total (explicado)

Un router solo "conoce" automáticamente las redes **directamente conectadas** a sus propias interfaces. Cualquier otra red le es **desconocida**: si un paquete va hacia una red que no está en su tabla, el router simplemente lo **descarta**. Por eso, en enrutamiento estático debes **configurar manualmente una ruta por cada red remota**.

Para que la comunicación sea total (todos pueden hablar con todos), revisa **router por router** y, para cada uno, agrega una ruta hacia **cada subred que NO esté conectada directamente** a él, sin olvidar los **enlaces WAN intermedios** (los /30 entre routers también son redes que hay que conocer para alcanzar destinos más lejanos).

Regla práctica de la ida y la vuelta: la conectividad requiere camino en **ambos sentidos**. No basta con que R1 sepa llegar a la red de R3; **R3 también debe saber cómo devolver la respuesta** a la red de R1. Si solo configuras una dirección, el `ping` sale pero nunca vuelve. Verifica siempre que cada destino tenga su ruta de regreso.

La tabla de enrutamiento: qué guarda y cómo se usa

Cada router mantiene su propia **tabla de enrutamiento** en memoria. Cuando llega un paquete, el router lee la **IP de destino** y busca en la tabla la mejor coincidencia para decidir por qué interfaz reenviarlo. Cada entrada incluye, como mínimo:

- **Red de destino + máscara** (ej. `10.20.0.0/25`).
- **Origen de la ruta:** conectada (`C`), local (`L`), estática (`S`) o aprendida por un protocolo (`R` RIP, `O` OSPF, `D` EIGRP).

- **Siguiente salto** (IP del próximo router) **o interfaz de salida**.
- **[Distancia administrativa / Métrica]:** criterios de preferencia cuando hay varias rutas.

```
Router# show ip route   (ejemplo de salida)
C    10.10.1.0/24   is directly connected, GigabitEthernet0/0
L    10.10.1.1/32   is directly connected, GigabitEthernet0/0
S    10.20.0.0/25   [1/0] via 172.16.10.2
S    192.168.50.0/28 [1/0] via 172.16.10.2
```

> **Regla de decisión: coincidencia más larga (longest prefix match)**
>
> Si un destino calza con varias rutas, el router siempre elige la de **prefijo más específico** (la máscara más larga). Solo si dos rutas tienen igual prefijo entra a comparar la **distancia administrativa** (qué tan "confiable" es el origen: conectada=0, estática=1, EIGRP=90, OSPF=110, RIP=120) y, si empata, la **métrica** del protocolo. En enrutamiento estático, tú llenas estas entradas **a mano** con el comando `ip route` .

Protocolos dinámicos y sus métricas

Protocolo	Tipo	Métrica que usa para elegir el camino
RIP	Vector distancia	Nº de saltos (hops). Elige el camino con **menos saltos** (máx. 15).
OSPF	Estado de enlace	Costo = 10^8 / BW(bps), sumado por enlace. Elige el de **menor costo total**.
EIGRP	Híbrido (DUAL)	Métrica compuesta basada en **ancho de banda + retardo** (y opcionalmente carga/ fiabilidad).

¿Cómo trabaja cada protocolo por dentro?

RIP (Routing Information Protocol). Funciona por **vector distancia**: cada router **no conoce la topología completa** de la red, solo sabe "en qué dirección y a qué distancia" está cada red, según lo que le cuentan sus vecinos. Periódicamente (cada 30 s) **envía toda su tabla de enrutamiento a los routers vecinos directos**; estos suman 1 salto y la propagan. Así la información se difunde salto a salto, como un rumor que se pasa de vecino en vecino (*"enrutamiento por rumores"*). Su métrica es simplemente el **conteo de saltos**, con un máximo de 15 (16 = red inalcanzable), lo que lo limita a redes pequeñas y lo hace de **convergencia lenta**.

OSPF (Open Shortest Path First). Funciona por **estado de enlace**: cada router **conoce el mapa completo** de la red. Para lograrlo, intercambia paquetes **LSA** (Link-State Advertisements) que describen sus enlaces y construye una base de datos topológica idéntica en todos los routers del área. Luego aplica el **algoritmo de Dijkstra (SPF)** sobre ese mapa para calcular el árbol de caminos más cortos hacia cada destino. Su métrica es el **costo**, inversamente proporcional al ancho de banda (`Costo = 10⁸ / BW`): a mayor ancho de banda, menor costo. Converge rápido y solo envía actualizaciones **cuando hay un cambio** en la red, no periódicamente.

EIGRP (Enhanced Interior Gateway Routing Protocol). Es **híbrido**: combina lo mejor del vector distancia con características de estado de enlace. Mantiene **tablas de vecinos y de topología**, y usa el algoritmo **DUAL** para calcular la mejor ruta (*successor*) y guardar una ruta de respaldo ya validada (*feasible successor*), lo que le permite **converger casi instantáneamente** ante una caída. Su métrica es **compuesta**: pondera principalmente **ancho de banda y retardo** acumulados del camino (y opcionalmente carga y fiabilidad), por lo

que toma decisiones más finas que RIP. Solo envía actualizaciones parciales y ante cambios, ahorrando ancho de banda.

OSPF: $\text{Costo}_{\text{enlace}} = 10^8 / \text{BW(bps)}$ (mínimo 1; se trunca a entero)

De dónde sale cada término (OSPF):

- **10^8 = ancho de banda de referencia** (100 Mbps en bps). Es un valor fijo que Cisco usa como "vara de medir".
- **BW(bps) = ancho de banda del enlace** en bits por segundo. Como está en el denominador, **a mayor ancho de banda, menor costo** (el enlace es "mejor").
- **Mínimo 1:** enlaces de ≥100 Mbps darían un costo <1, pero OSPF no permite costo 0, así que lo fija en 1. El resultado se **trunca a entero** (se descartan los decimales).
- El costo de una **ruta completa** es la **suma** de los costos de cada enlace que la compone.

EIGRP (forma simplificada del curso):

$$\text{Costo} = 256 \cdot \left[\left(10^7 / \text{BW}_{\text{peor caso}}(\text{Kbps})\right) + \left(\Sigma\text{Delay}(\mu s) / 10 \right) \right]$$

De dónde sale cada término (EIGRP):

- **$\text{BW}_{\text{peor caso}}$ (Kbps):** el **ancho de banda más bajo** (el enlace más lento) de todo el camino, porque una cadena es tan rápida como su eslabón más débil. Va en Kbps.
- **10^7 / $\text{BW}_{\text{peor caso}}$:** componente de **ancho de banda** (mismo principio que OSPF: más BW → menor valor).
- **ΣDelay(µs):** la **suma de los retardos** de todos los enlaces del camino, en microsegundos. Se divide entre 10 para convertir a "decenas de µs" (unidad interna de EIGRP).
- **× 256:** factor de escala histórico (IGRP usaba números más chicos; EIGRP los multiplica por 256).

La gran diferencia entre ambas

OSPF solo mira **ancho de banda**. EIGRP mira **ancho de banda + retardo** a la vez, por eso su métrica es "compuesta" y puede tomar decisiones más finas entre caminos parecidos.

El gran malentendido: RIP vs OSPF/EIGRP

RIP solo cuenta saltos: puede elegir un enlace **lento** (ej. un serial T1) solo porque tiene menos saltos. OSPF y EIGRP sí consideran el **ancho de banda**, por lo que suelen elegir caminos más rápidos aunque tengan más saltos. ¡Este contraste es típico de evaluación!

7. Capa de Transporte

Objetivo de la capa de transporte

La capa de transporte es la encargada de establecer una **comunicación lógica extremo a extremo (end-to-end)** entre **procesos/aplicaciones** que se ejecutan en hosts distintos. Mientras la capa de red (IP) se

preocupa de llevar paquetes de un **host a otro**, la capa de transporte da el paso siguiente y crucial: entregar los datos a la **aplicación correcta dentro de ese host**. Lo hace mediante dos mecanismos fundamentales:

- **Multiplexación / demultiplexación:** usa **números de puerto** para que muchos programas compartan la misma conexión de red y los datos lleguen a la app correcta (ej. el navegador en el 443, el correo en el 25).
- **Segmentación:** divide los datos de la aplicación en unidades manejables (**segmentos**) y los reensambla en el destino.

Propósito fundamental de TCP y UDP

Sobre esa base, existen dos protocolos con filosofías opuestas. El **propósito de TCP** es ofrecer un transporte **confiable y ordenado**: garantizar que **todos** los datos lleguen, **completos** y en el **orden correcto**, aunque eso cueste más tiempo y recursos. El **propósito de UDP** es ofrecer un transporte **rápido y ligero**: enviar los datos con el mínimo retardo y sobrecarga posibles, sacrificando las garantías de entrega.

Lo que AMBOS deben cumplir sí o sí

Más allá de sus diferencias, TCP y UDP son protocolos de transporte y por tanto **los dos están obligados** a:

- **Comunicación extremo a extremo (proceso a proceso):** conectar la aplicación origen con la aplicación destino, no solo host con host.
- **Multiplexación con puertos:** ambos identifican la aplicación mediante un **número de puerto** (origen y destino), formando el **socket**.
- **Detección de errores básica:** los dos incluyen un campo **checksum** en su cabecera para detectar datos corruptos (la diferencia es qué hacen luego: TCP retransmite, UDP simplemente descarta el segmento dañado).

En resumen: **ambos deben entregar los datos al proceso correcto y verificar su integridad**; lo que cambia es **qué garantías adicionales** ofrece cada uno.

TCP vs UDP

Característica	TCP	UDP
Conexión	Orientado a conexión (saludo de 3 vías / 3-way handshake)	No orientado a conexión
Fiabilidad	Confiable: ACKs + retransmisiones	No confiable: best-effort, sin retransmisión
Orden	Entrega ordenada (números de secuencia)	Sin orden garantizado
Velocidad / overhead	Más lento, mayor overhead	Rápido, bajo overhead, baja latencia
Control	Control de flujo y de congestión	Ninguno
Uso típico	Web, correo, transferencia de archivos	Streaming, VoIP, juegos, DNS

Ojo: en UDP el orden lo resuelve la aplicación, no el transporte

En streaming, VoIP o juegos, UDP **no** ordena ni recupera paquetes. Cuando esas aplicaciones necesitan orden o sincronía, es la **capa de aplicación** la que se encarga: protocolos como **RTP** añaden sus propios números de secuencia y marcas de tiempo, y un **buffer de jitter** reordena y suaviza la reproducción. Es decir, la responsabilidad del orden **sube a la aplicación** porque el transporte (UDP) deliberadamente no lo hace, a cambio de mínima latencia.

Establecimiento de conexión TCP: el saludo de 3 vías (3-way handshake)

Antes de enviar datos, TCP **establece la conexión** y **sincroniza los números de secuencia** de ambos extremos mediante tres mensajes que usan los bits de control (flags) **SYN** y **ACK**:

1. **SYN** → El cliente envía un segmento con el flag **SYN** y un número de secuencia inicial (ej. `seq = x`). Pide abrir la conexión.
2. ← **SYN-ACK** El servidor responde con **SYN + ACK**: confirma el del cliente (`ack = x+1`) y envía su propio número de secuencia (`seq = y`).
3. **ACK** → El cliente confirma el del servidor (`ack = y+1`). La conexión queda **establecida** y comienza el envío de datos.

```
Cliente                          Servidor
  | ------- SYN (seq=x) --------->  |
  | <--- SYN-ACK (seq=y, ack=x+1) - |
  | ------- ACK (ack=y+1) -------->  |
  |        conexión establecida      |
```

¿Para qué sirve el handshake?

Garantiza que **ambos extremos están activos y de acuerdo** en iniciar, y **acuerda los números de secuencia** que permitirán detectar pérdidas, evitar duplicados y entregar los datos en orden. El cierre de la conexión usa un proceso análogo con el flag **FIN** (normalmente de 4 pasos). Este saludo es justamente parte del **overhead** que UDP se ahorra.

Control de flujo (¡no confundir con control de congestión!)

El **control de flujo** evita que el emisor sature al **receptor** (regula cuánto enviar antes de recibir un ACK). El **control de congestión** evita saturar la **red intermedia**. TCP implementa ambos; UDP, ninguno.

Sockets y puertos

Un **socket** es la combinación **Dirección IP + Número de puerto**. Una sesión queda identificada de forma única por la **tupla de 4 elementos**:

```
[ IP origen · Puerto origen · IP destino · Puerto destino ]
```

Por eso un servidor web atiende a miles de clientes **en el mismo puerto 443**: lo que distingue cada conexión es la combinación completa (sobre todo la IP y el puerto de origen del cliente).

Puertos predeterminados frecuentes

Servicio	Puerto	Transporte	Descripción
FTP	20 / 21	TCP	Transferencia de archivos (20 datos / 21 control)
SSH	22	TCP	Administración remota segura
SMTP	25	TCP	Envío de correo entre servidores
DNS	53	UDP / TCP	Traduce nombres de dominio a IP
HTTP	80	TCP	Web (sin cifrar)
HTTPS	443	TCP	Web cifrada

EVALUACIÓN PRINCIPAL

8. Solemne 2 — Ejercicios Resueltos

Guía de ejercicios de refuerzo · CIT2114 · con soluciones verificadas paso a paso

Para el estudiante

Intenta resolver cada ejercicio **por tu cuenta primero** y luego compara con la solución. Todas las respuestas de esta sección fueron **verificadas con cálculo programático**. Lo importante no es la respuesta final, sino entender el **procedimiento**.

Sección I — Selección múltiple

1. Sockets y puertos de capa de transporte

Servidor web `190.22.45.10` atiende HTTPS. Tres clientes concurrentes se conectan. ¿Qué describe correctamente el comportamiento de los sockets?

A) Usaría tres puertos destino distintos (443, 444, 445).

B) Cada sesión queda identificada por la tupla [IP origen, Puerto origen, IP destino, Puerto destino 443].

C) Requiere las MAC de los clientes en el identificador del puerto.

D) Cierra temporalmente la escucha en el 443 hasta procesar al primer cliente.

Respuesta: B. El servidor mantiene **un único puerto de escucha (443)** para todos. Lo que diferencia cada conexión es el **socket completo de 4 elementos**; como los tres clientes tienen IP y/o puerto de origen distintos, cada sesión es única. A es falsa (no se abren puertos 443/444/445), C es falsa (la MAC es de capa 2, no del identificador de transporte) y D describe un comportamiento inexistente.

2. IPv4 y VLSM: dirección 172.24.96.142/22

Hallar la dirección de red y la última dirección útil.

Respuesta: A) 172.24.96.0 y 172.24.99.254

Desarrollo: /22 = `255.255.252.0`. El bloque está en el **3er octeto**: incremento = 256 − 252 = **4**. Los múltiplos de 4 son 0, 4, …, 92, **96**, 100… Como 96 es múltiplo de 4, la red empieza en **172.24.96.0**.

- Dirección de red: **172.24.96.0**
- Broadcast: la dirección anterior a la siguiente red (172.24.100.0) → **172.24.99.255**
- Última IP útil = broadcast − 1 = **172.24.99.254**

3. Máscara para 2046 hosts útiles

Respuesta: B) 255.255.248.0 (/21)

Desarrollo: buscamos x tal que 2^x − 2 ≥ 2046. Con **x = 11**: $2^{11} - 2 = 2048 - 2 =$ **2046** exacto. Prefijo = 32 − 11 = **/21** → máscara `255.255.248.0`.

Verificación de descartes: /20 da 4094 (sobra), /22 da 1022 (insuficiente), /23 da 510 (insuficiente). El /21 es el único óptimo y exacto.

4. Simplificación IPv6: 2001:0DB8:0000:4A00:0000:0000:0000:001F

Respuesta: A) ipv6 address 2001:db8:0:4a00::1f

Desarrollo: quitar ceros iniciales → `2001:db8:0:4a00:0:0:0:1f`; comprimir la secuencia más larga de ceros (los 3 hextetos finales antes de 1f) con `::`.

B es inválida: usa `::` **dos veces** (ambigua). C es correcta pero **no está comprimida** (no usa `::`). D deja ceros sin simplificar. La forma óptima es A.

Sección II — Desarrollo técnico

Ejercicio 1: Diseño VLSM sobre 192.168.40.0/24

Institución de salud. Ordenar de mayor a menor y completar la tabla sin dejar espacio muerto. Requerimientos: A=62, B=28, C=14, D=12, E=2 hosts. Ya vienen en orden descendente.

Subred (uso)	Hosts	Red (CIDR)	Máscara decimal	1ª IP útil	Última IP útil	Broadcast
A · Red Médica	62	192.168.40.0/26	255.255.255.192	192.168.40.1	192.168.40.62	192.168.40.63
B · Administración	28	192.168.40.64/27	255.255.255.224	192.168.40.65	192.168.40.94	192.168.40.95
C · Laboratorios	14	192.168.40.96/28	255.255.255.240	192.168.40.97	192.168.40.110	192.168.40.111
D · Consultas Ext.	12	192.168.40.112/28	255.255.255.240	192.168.40.113	192.168.40.126	192.168.40.127
E · Enlace WAN	2	192.168.40.128/30	255.255.255.252	192.168.40.129	192.168.40.130	192.168.40.131

Cómo se llegó a cada prefijo

A: 2^6−2=62 ≥ 62 → /26. B: 2^5−2=30 ≥ 28 → /27. C: 2^4−2=14 ≥ 14 → /28. D: 2^4−2=14 ≥ 12 → /28. E: 2^2−2=2 ≥ 2 → /30. Cada red parte donde terminó (broadcast+1) la anterior: 0 → 64 → 96 → 112 → 128.

Dato interesante: aunque C (14) y D (12) requieren distinta cantidad de hosts, ambas caben en un /28, porque /28 es el bloque más chico que satisface a las dos.

Ejercicio 2: Métricas de enrutamiento y selección de camino

Topología con 4 routers. Caminos de R1 a R4:

- **Ruta A (superior):** R1-R2 (FastE, 100 Mbps, 100 µs) + R2-R4 (FastE, 100 Mbps, 100 µs). **2 saltos.**
- **Ruta B (central):** R1-R4 directo (serial T1, 1.544 Mbps, 20.000 µs). **1 salto.**
- **Ruta C (inferior):** R1-R3 (Gigabit, 1000 Mbps, 10 µs) + R3-R4 (banda estrecha, 64 Kbps, 20.000 µs). **2 saltos.**

a) RIP — métrica = número de saltos

RIP elige la **Ruta B** (R1→R4 directo), con métrica = **1 salto**. A y C tienen 2 saltos cada una.

Atención

RIP elige el enlace serial T1 lento **solo porque tiene menos saltos**, ignorando que su ancho de banda (1.544 Mbps) es muchísimo menor que los 100 Mbps de la Ruta A. Es la principal debilidad de RIP.

b) OSPF — Costo = 10^8 / BW(bps), sumado por enlace

Ruta	Cálculo por enlace	Costo total
A	(10^8/100M = 1) + (10^8/100M = 1)	**2**
B	10^8/1.544M = 64,77 → 64	**64**
C	(10^8/1000M = 0,1 → 1) + (10^8/64K = 1562,5 → 1562)	**1563**

OSPF elige la **Ruta A**, con costo total = **2** (el menor de los tres).

Nota: el costo mínimo de un enlace en OSPF es 1, por eso el enlace Gigabit (0,1) se redondea a 1. Cisco trunca el resultado a entero.

c) EIGRP — fórmula compuesta simplificada

$$\text{Costo} = 256 \cdot \left[\left(10^7 / \text{BW}_{\text{peor caso}}(\text{Kbps})\right) + \left(\Sigma\text{Delay}(\mu s) / 10\right) \right]$$

Ruta	BW peor caso	ΣDelay	Cálculo	Métrica
A	100.000 Kbps	200 µs	256·[(10^7/100000) + (200/10)] = 256·[100 + 20] = 256·120	**30.720**
B	1.544 Kbps	20.000 µs	256·[(10^7/1544) + (20000/10)] = 256·[6476,68 + 2000] = 256·8476,68	**≈ 2.170.031**

EIGRP (algoritmo DUAL) elige la **Ruta A**: su métrica (30.720) es muchísimo menor que la de la Ruta B (≈ 2.170.031). La Ruta A es la óptima.

Conclusión comparativa (lo que debes recordar)

Misma topología, tres decisiones distintas: **RIP** → **Ruta B** (solo cuenta saltos), **OSPF** → **Ruta A** (mira ancho de banda), **EIGRP** → **Ruta A** (ancho de banda + retardo). Esto demuestra por qué los protocolos modernos eligen mejores caminos que RIP.

Sección III — Caso de estudio: telemetría en UCI (TCP vs UDP)

Contexto: hospital monitorea signos vitales de pacientes críticos en tiempo real. Requiere **latencia < 20 ms** para alertas inmediatas, pero la pérdida o desorden severo de paquetes podría provocar falsos negativos fatales.

Respuesta modelo (fundamentación técnica)

1. Análisis del requerimiento. Hay dos exigencias en tensión: *latencia ultrabaja* (favorece UDP) y *no perder información crítica* (favorece TCP). Por eso ninguno de los dos protocolos clásicos, por sí solo, es ideal.

2. Por qué TCP puro no es adecuado. TCP es confiable y ordenado, pero su **retransmisión + control de congestión** introduce retardos variables y **head-of-line blocking**: un paquete perdido detiene la entrega de los siguientes hasta retransmitirlo. En telemetría continua, un dato de signos vitales retransmitido suele llegar **demasiado tarde** para ser útil, y el handshake de 3 vías y los ACKs añaden latencia que puede superar los 20 ms.

3. Por qué UDP puro tampoco basta. UDP ofrece la baja latencia y el bajo overhead necesarios (sin handshake, sin retransmisión, sin HOL blocking), pero **no garantiza entrega ni orden**: una ráfaga de pérdidas podría ocultar una alarma de paro cardiorrespiratorio.

4. Solución arquitectónica propuesta (enfoque híbrido). Usar **UDP como base de transporte** (para preservar la latencia) y añadir **fiabilidad selectiva en la capa de aplicación**:

- **Números de secuencia** a nivel de aplicación para detectar pérdidas y reordenar.
- **Retransmisión selectiva solo de paquetes críticos** (alarmas), no del flujo continuo de telemetría.
- **FEC (Forward Error Correction):** enviar información redundante para reconstruir paquetes perdidos sin esperar retransmisión.
- **Doble canal:** un flujo UDP para la telemetría continua de baja latencia, y un canal confiable (TCP o ACK de aplicación) para **confirmar las alarmas críticas**.
- **Heartbeat / keepalive** para detectar caída del enlace o del sensor rápidamente.

5. El rol de la red (QoS). La elección del protocolo no basta: la red debe **garantizar** la baja latencia. Se aplica **Calidad de Servicio (QoS)** para **priorizar** el tráfico de telemetría sobre el tráfico común, reservando ancho de banda y minimizando el **jitter** (variación del retardo) y la pérdida en los switches y routers intermedios.

> **Síntesis**
>
> La mejor arquitectura combina **la velocidad de UDP** con **fiabilidad selectiva en la aplicación**, priorizando que las **alarmas críticas lleguen sí o sí** mientras el flujo continuo se entrega con mínima latencia. Tecnologías reales que aplican esta idea: RTP/RTCP, DTLS o QUIC sobre UDP.

> **Recomendación final concreta**
>
> **No usar TCP puro** (sus retransmisiones y head-of-line blocking rompen el límite de 20 ms). **Adoptar UDP como transporte base** + fiabilidad selectiva en la aplicación (secuencia, FEC, retransmisión solo de alarmas) + un **canal confiable separado para confirmar las alertas críticas** + **QoS** en la red. Así se satisface **simultáneamente** la latencia ultrabaja y la garantía de que ninguna alarma vital se pierda — los dos requisitos que el sistema debe cumplir sí o sí.

A RESOLVER POR TU CUENTA

9. Ejercicios de Práctica

Mismos formatos que la Solemne 2, con datos distintos · sin respuestas

Cómo aprovechar esta sección

Estos ejercicios siguen exactamente la estructura de los resueltos en la Sección 8, pero con **valores diferentes**. Resuélvelos sin mirar las soluciones y luego contrasta tu método con el de la Sección 8. Si tu procedimiento es correcto, el resultado lo será. **No se incluyen respuestas a propósito**, para que practiques de verdad.

Sección I — Selección múltiple

P1. Sockets y puertos

Un servidor de correo en `200.14.20.5` recibe conexiones SMTP de cuatro clientes distintos al mismo tiempo. ¿Qué permite al servidor distinguir cada sesión usando un solo puerto de escucha?

A) Asigna un puerto destino distinto (25, 26, 27, 28) a cada cliente.

B) Cada sesión se identifica por la tupla [IP origen, Puerto origen, IP destino, Puerto destino].

C) Incorpora la MAC de cada cliente al número de puerto.

D) Necesita una IP pública diferente por cada cliente.

P2. IPv4 / VLSM

Dada la dirección de host `10.45.130.200/21`, determina la **dirección de red** y la **última dirección útil** de su subred.

P3. Diseño de máscara

¿Cuál es el prefijo CIDR y la máscara decimal **óptima** para una subred que debe alojar al menos **500 hosts útiles**?

P4. Simplificación IPv6

Simplifica al máximo la dirección, aplicando las reglas de RFC 4291:

`2001:0DB8:0000:0000:0000:00FF:0000:0001`

Sección II — Desarrollo técnico

Ejercicio 1: Diseño VLSM

A partir del bloque `172.16.20.0/24`, diseña un esquema VLSM ordenado de mayor a menor y completa la tabla. Requerimientos: **Ventas 50, Bodega 25, Gerencia 10, Soporte 5, Enlace WAN 2**.

Subred	Hosts	Red (CIDR)	Máscara	1ª IP útil	Última IP útil	Broadcast
Ventas	50					
Bodega	25					
Gerencia	10					
Soporte	5					
Enlace WAN	2					

Ejercicio 2: Métricas de enrutamiento

Datos críticos viajan de R1 a R5 por dos caminos posibles:

- **Camino X:** R1-R2 (Gigabit, 1000 Mbps, 10 µs) + R2-R5 (10 Mbps, 1000 µs). **2 saltos.**
- **Camino Y:** R1-R3 + R3-R4 + R4-R5, los tres Fast Ethernet (100 Mbps, 100 µs c/u). **3 saltos.**

Determina qué camino elige cada protocolo, justificando con la métrica:

1. **RIP** (por número de saltos).
2. **OSPF** (Costo = 10^8/BW por enlace, sumado).
3. **EIGRP** (fórmula simplificada del curso, calcula ambos caminos).

Ejercicio 3: Caso de estudio TCP/UDP

Una plataforma ofrece **videojuegos multijugador en línea** en tiempo real y, además, un módulo de **transacciones de compra** dentro del juego. Analiza qué protocolo de transporte conviene para **cada flujo** (el del juego y el de las compras), justificando según latencia, fiabilidad y orden. Propón una arquitectura que use ambos protocolos donde corresponda.

10. Formulario y Referencia Rápida

Fórmulas esenciales

Hosts útiles = 2^(bits host) − 2 · Subredes = 2^(bits robados) · Prefijo = 32 − bits host

Incremento = 256 − (octeto de la máscara) · Broadcast = (red siguiente) − 1

Shannon: $C = B \cdot \log_2(1 + SNR)$ · $SNR_{abs} = 10^{(dB/10)}$

Fibra: $NA = \sqrt{(n_1^2 - n_2^2)}$ · OSPF: Costo = 10^8 / BW(bps)

EIGRP (simpl.): $256 \cdot [(10^7 / BW_{Kbps}) + (\Sigma Delay_{\mu s} / 10)]$

Qué significa cada símbolo del formulario

C = capacidad (bps) · **B** = ancho de banda (Hz) · **SNR** = relación señal/ruido (valor absoluto) · **NA** = apertura numérica · **n_1 / n_2** = índice de refracción del núcleo / revestimiento · **BW** = ancho de banda del enlace · **ΣDelay** = suma de los retardos del camino (µs) · **bits host** = 32 − prefijo · **bits robados** = bits tomados para crear subredes.

Tabla de prefijos CIDR (4º octeto)

Prefijo	Máscara	Hosts útiles	Incremento
/25	255.255.255.128	126	128
/26	255.255.255.192	62	64
/27	255.255.255.224	30	32
/28	255.255.255.240	14	16
/29	255.255.255.248	6	8
/30	255.255.255.252	2	4

Potencias de 2 (memorízalas)

2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
2	4	8	16	32	64	128	256	512	1024	2048

Checklist de errores comunes a evitar

No te equivoques en la evaluación

- En VLSM, **ordenar de mayor a menor** SIEMPRE antes de asignar.
- Restar **2** (red + broadcast) al contar hosts útiles, salvo casos especiales.
- En IPv6, usar `::` **una sola vez** y comprimir la secuencia más larga de ceros.
- En Shannon, convertir dB a SNR absoluto antes de calcular.
- El router decide con la **tabla de enrutamiento**, no con ARP/MAC/DNS.
- En cada salto cambia la **MAC**, la **IP se mantiene**.
- RIP cuenta saltos; OSPF/EIGRP miran el ancho de banda → distinto camino elegido.

¡Éxito en la Solemne 2!

Repasa especialmente la Sección 8. Practica los cálculos de VLSM y métricas con lápiz y papel hasta que el procedimiento te salga automático. Entender el *por qué* de cada paso es lo que realmente marca la diferencia.

Material de estudio elaborado a partir de las guías oficiales del curso Redes de Datos (CIT2114) · Universidad Diego Portales · Todos los cálculos fueron verificados.