

Fundamentos de Redes de Comunicaciones

Volumen 2

Luciana B. Falcon

3 de agosto de 2025

Índice

1. Nivel de Enlace y Redes LAN	2
1.1. Protocolo ARP	2
2. Ruteo Interno - RIP - OSPF	5
2.1. Ruteo	5
2.2. Protocolos de ruteo interno	7
2.3. Algoritmo de Distancia Vectorial	8
2.3.1. Protocolo RIP	11
2.4. Algoritmo de Link State	11
2.4.1. Protocolo OSFP	11
3. Arquitectura	12
3.1. Inicios y gestión de Internet	12
3.2. Interconexión	16
3.3. Tipos de redes	17
4. Redes Definidas por Software	23
4.1. Arquitectura de un Router	23
4.2. SDN	24
4.3. Protocolo OpenFlow	27
4.4. P4	30
4.5. Open Compute Project (OCP)	30
5. Nivel Físico	33
5.1. Medio físicos	33
5.2. Modulación	35
5.3. Multiplexación	38
5.4. Teoremas de Nyquist y Shannon	39

1. Nivel de Enlace y Redes LAN

Nivel de Red (IP) → Nivel de Enlace (Ethernet) → Nivel Físico (cable, fibra, etc.)

Consideraciones para el Despacho Local:

Cuando el origen y el destino se encuentran en la misma red de acceso, el proceso para enviar datos tiene algunas particularidades importantes. La estación de origen necesita conocer la dirección del destino o, en su defecto, la dirección del siguiente salto a nivel de enlace.

Si la red utiliza Ethernet, esta dirección corresponde a una *MAC address* (dirección física). En caso de no conocer esta dirección, la estación origen realiza una consulta al protocolo ARP (Address Resolution Protocol) para obtenerla (detalle que continuará).

El datagrama IP se transmite encapsulado dentro de una trama del nivel de enlace, como puede ser una trama Ethernet. Durante todo este proceso, las direcciones IP, que pertenecen al nivel de red, no cambian.

Por lo tanto, este proceso corresponde principalmente al nivel de red, que gestiona las direcciones IP y el encaminamiento lógico, trabajando en conjunto con el nivel de enlace, que se encarga de la entrega física dentro de la red local.

1.1. Protocolo ARP

El *Address Resolution Protocol* (ARP) está definido en las RFC 826, 5227 y 5494. Su función principal es realizar el mapeo entre direcciones lógicas (direcciones IP) y físicas (direcciones MAC), ya sea de forma estática o dinámica.

Por ejemplo, ARP se utiliza para resolver una dirección IP y obtener la dirección MAC correspondiente dentro de una red local, permitiendo así la comunicación a nivel de enlace.

Existen otros protocolos relacionados que cumplen funciones similares en diferentes contextos:

- **iARP (Inverse ARP):** Mapea un identificador DLCI (Data Link Connection Identifier) a una dirección IP.
- **RARP (Reverse ARP):** Realiza la operación inversa a ARP, es decir, obtiene una dirección IP a partir de una dirección MAC.

Funcionamiento

Cuando un dispositivo necesita conocer la dirección MAC asociada a una dirección IP, envía un mensaje ARP *Request* en modo broadcast, es decir, dirigido a toda

la red local.

El dispositivo que posee la dirección IP solicitada responde con un mensaje ARP *Reply* en modo unicast, dirigido exclusivamente al solicitante. De esta manera, solo quien realizó la consulta recibe la información requerida.

ARP opera en el nivel de enlace de datos para facilitar la comunicación entre dispositivos en una red local, traduciendo direcciones IP (nivel de red) a direcciones MAC (nivel de enlace).

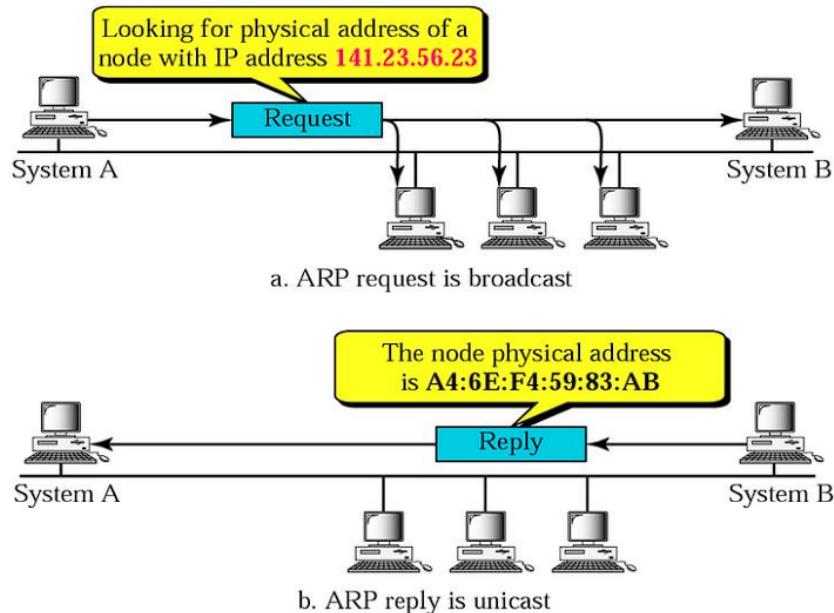


Figura 1: ARP en redes LAN-Ethernet.

0	8	16	bit 31		
Hardware type		Protocol type			
HW address length	Prot address length	Operation			
Sender Hardware address					
(cont.)		Sender protocol address			
(cont.)		Target hardware address			
(cont.)					
Target protocol address					

Hardware address Type : Tipo de protocolo del nivel 2. (Ethernet=1)

Protocol address Type : Tipo de protocolo de nivel 3 ($IP = 0800_{16}$)

Operation : 1=request, 2=response.

Figura 2: Formatos de las tramas ARP.

```
> Frame 347: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NP
└╼ Ethernet II, Src: Routerboardc_bb:85:f5 (d4:ca:6d:bb:85:f5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    > Source: Routerboardc_bb:85:f5 (d4:ca:6d:bb:85:f5)
    Type: ARP (0x0806)
    Padding: 0000000000000000000000000000000000000000000000000000000000000000
    └╼ Address Resolution Protocol (request)
        Hardware type: Ethernet (1)
        Protocol type: IPv4 (0x0800)
        Hardware size: 6
        Protocol size: 4
        Opcode: request (1)
        Sender MAC address: Routerboardc_bb:85:f5 (d4:ca:6d:bb:85:f5)
        Sender IP address: 192.168.150.1 (192.168.150.1)
        Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
        Target IP address: 192.168.150.115 (192.168.150.115)
```

Figura 3: Analizando el ARP Request en la LAN.

```
> Frame 364: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NP
└╼ Ethernet II, Src: Intel_f8:26:90 (10:3d:1c:f8:26:90), Dst: Routerboardc_bb:85:f5 (d4:ca:6d:b
    > Destination: Routerboardc_bb:85:f5 (d4:ca:6d:bb:85:f5)
    > Source: Intel_f8:26:90 (10:3d:1c:f8:26:90)
    Type: ARP (0x0806)
    └╼ Address Resolution Protocol (reply)
        Hardware type: Ethernet (1)
        Protocol type: IPv4 (0x0800)
        Hardware size: 6
        Protocol size: 4
        Opcode: reply (2)
        Sender MAC address: Intel_f8:26:90 (10:3d:1c:f8:26:90)
        Sender IP address: 192.168.150.118 (192.168.150.118)
        Target MAC address: Routerboardc_bb:85:f5 (d4:ca:6d:bb:85:f5)
        Target IP address: 192.168.150.1 (192.168.150.1)
```

Figura 4: Analizando el ARP Reply en la LAN.

2. Ruteo Interno - RIP - OSPF

2.1. Ruteo

Ruteo vs. “Forwarding”

Forwarding es un proceso local y rápido realizado por cada router. Selecciona un port de salida en función de destino y tabla de ruteo.

En cambio, el ruteo es un proceso global y más complejo que puede usar algoritmos distribuidos. Es decir es el proceso mediante el cual se construye la tabla de ruteo.

Desafíos del proceso de ruteo

El ruteo necesita información sobre toda la red (como la topología, los costos de los enlaces, el estado de congestión, etc.). Este proceso puede ser lento (*time-consuming*), ya que la red es dinámica (cambia constantemente), lo que hace que sea difícil recolectar y mantener actualizada esta información.

Cuestiones clave a resolver:

- Consistencia: que todos los routers tengan una visión coherente y actualizada de la información de ruteo.
- Completitud: que la información incluya todos los destinos posibles dentro de la red.
- Escalabilidad.

Consistencia

Para garantizar una conectividad efectiva entre cualquier par de origen y destino en la red, las decisiones de forwarding local tomadas por los routers deben estar basadas en información de ruteo consistente.

Si los estados de ruteo no están sincronizados entre los routers (es decir, si hay inconsistencias), la red no habrá convergido a un estado estable. Esta falta de convergencia puede generar comportamientos indeseados como loops de ruteo, en los que los paquetes circulan indefinidamente entre routers sin llegar a destino.

Básicamente significa que los routers comparten una visión sincronizada de la red. Si esta información no coincide entre ellos, la red no ha convergido y pueden aparecer problemas como loops de ruteo o pérdida de paquetes.

Completitud

La completitud en los algoritmos de ruteo implica que tanto la red en su conjunto como cada nodo individual deben disponer de suficiente información para poder

hallar rutas hacia todos los destinos posibles.

Cuanto mayor sea la información disponible localmente en cada router, más rápido convergen los algoritmos de ruteo. Sin embargo, esta ventaja viene acompañada de una mayor necesidad de recolectar y almacenar información en cada nodo, lo que puede afectar negativamente la escalabilidad del sistema.

A medida que la red crece, el volumen de información requerido también lo hace, lo cual puede generar una sobrecarga en los recursos de procesamiento y memoria de los routers.

Ruteo Estático vs Dinámico

En el *Ruteo estático* las rutas se configuran manualmente por el administrador. Brinda mayor control sobre el camino que toman los paquetes. No es escalable, ya que requiere intervención en cada router. Es lento para adaptarse a los cambios en la topología de red (por ejemplo, fallas o nuevos enlaces).

En cambio en el *Ruteo dinámico* los routers utilizan protocolos de ruteo para calcular automáticamente los caminos óptimos (como RIP, OSPF). Permite una adaptación rápida a cambios en la red. Es escalable, adecuado para redes grandes. Consumo más CPU, ancho de banda (BW) y memoria. El debugging o resolución de problemas es más complejo, debido al dinamismo y cantidad de información intercambiada.

- ✓ Decisiones de “forwarding” local deben llevar a una conectividad entre cualquier origen y destino de la red
- ✓ Si los estados son inconsistentes la red no habrá convergido a un estado estable
- ✓ La inconsistencia genera loops
- ✓ La red como un todo y cada nodo deben tener la información suficiente para poder hallar todos los caminos
- ✓ Cuanto mayor sea la información disponible localmente mayor es la rapidez con que convergen los algoritmos de ruteo
- ✓ Eso significa que se debe recolectar mayor información en cada nodo
- ✓ Esto también limita la escalabilidad del algoritmo

Figura 5: Consistencia: todos los routers deben tener la misma información de ruteo.

Figura 6: Completitud: la información debe incluir todos los destinos posibles.

Estático	Dinámico
<ul style="list-style-type: none"> ● Configuración manual de las tablas ● Mayor control ● No es escalable ● Lento para adaptarse a los cambios en la red 	<ul style="list-style-type: none"> ● Se obtienen caminos óptimos ● Rápida adaptación a los cambios ● Escalables ● Consumo de CPU, BW y Memoria ● “Debugging” complejo

Figura 7: Ruteo Estático vs Dinámico.

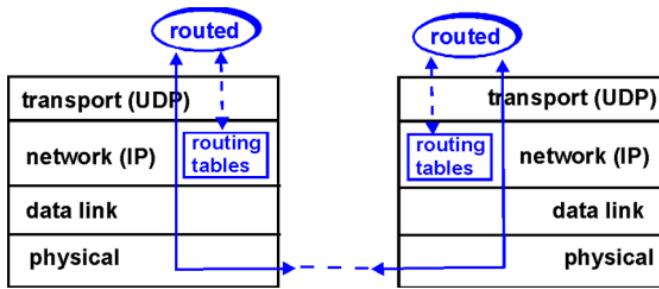


Figura 8: Esquema de ruteo dinámico.

Modelo de Ruteo en Internet

Internet utiliza un modelo de **ruteo dinámico y jerárquico**, organizado a partir de **Sistemas Autónomos (SA)**.

El Sistema Autónomo (SA) es un conjunto de routers bajo una misma administración o política de ruteo. Cada SA se identifica mediante un número único (ASN) y toma decisiones internas de ruteo.

Tipos de ruteo:

- **Ruteo interno (Intra-AS):**
Se realiza dentro de un SA mediante protocolos IGP (Interior Gateway Protocol), como RIP y OSPF.
- **Ruteo externo (Inter-AS):**
Se realiza entre SA distintos, mediante protocolos EGP (Exterior Gateway Protocol), como BGP-4.

2.2. Protocolos de ruteo interno

Algoritmos de Ruteo Dinámico

En el ruteo dinámico se utilizan algoritmos distribuidos para calcular caminos óptimos en la red. Existen dos enfoques principales: el Vector de Distancia (Distance Vector) y el Estado de Enlace (Link-State).

- ✓ **Distancia Vectorial**
 - Los nodos intercambian tablas con sus vecinos

- ✓ **“Link-state”:** información basada en los enlaces
 - Obtiene un mapa completo de la red
 - Encuentran el camino óptimo y determinan el próximo salto localmente(Algoritmo de Dijkstra).

Figura 9: Algoritmos Distancia Vectorial y Link-state.

2.3. Algoritmo de Distancia Vectorial

Utiliza un algoritmo distribuido, típicamente Bellman-Ford. Cada router intercambia información de ruteo únicamente con sus vecinos directamente conectados. El intercambio de información es periódico, y no ocurre inmediatamente ante cada cambio. De esta forma, cada router construye su tabla de ruteo a partir de un conocimiento parcial de la red, sin contar con una visión global de la topología.

Entre las principales consecuencias de este enfoque se encuentra una convergencia lenta, ya que puede demorar en propagarse un cambio a toda la red. Además, existe la posibilidad de aparición de lazos de ruteo (routing loops), especialmente durante el proceso de convergencia. Por otro lado, este enfoque requiere pocos recursos, ya que el consumo de CPU, ancho de banda y memoria es bajo en comparación con otros algoritmos más complejos.

- ✓ Algoritmo distribuido – Bellman-Ford
- ✓ Cada nodo intercambia información con sus vecinos (nodos directamente conectados)
- ✓ Intercambio de información periódico.
- ✓ Mapa incompleto de la red
- ✓ Consecuencias:
 - Convergencia Lenta
 - Existencia de Lazos
 - Pocos recursos de procesamiento y memoria

Figura 10: Distancia vectorial.

Algoritmo Vectorial en funcionamiento

Como concepto general, cada router mantiene una tabla de ruteo que indica:

- Las redes de destino que conoce.
- La métrica asociada (número de saltos).
- A qué vecino debe enviar los paquetes (próximo salto) para alcanzar cada red.

Los routers intercambian periódicamente esta información con sus vecinos directos y actualizan sus tablas si encuentran rutas más eficientes.

El algoritmo *Distance Vector Routing Algorithm* se basa en los siguientes principios:

1. Cada router mantiene una **tabla de ruteo** con:
 - La lista de destinos posibles.
 - La métrica (generalmente, número de saltos) hacia cada destino.
 - El próximo salto (vecino) para alcanzar ese destino.

2. Los routers **comparten periódicamente sus tablas de ruteo** con sus vecinos directos.
3. Al recibir una tabla, un router **actualiza su propia tabla** si descubre una ruta más eficiente (menor métrica).
4. Este proceso se repite continuamente hasta que todas las rutas óptimas se propagan por la red (convergencia).

Ejemplo:

Supongamos una red con tres routers conectados en línea:

$$A \longleftrightarrow B \longleftrightarrow C$$

Inicialmente:

- A conoce solo a B - Router A tiene dos interfaces:
S0 → conectado a la red 162.11.5.0
S1 → conectado a Router B a través de la red 162.11.8.0
- B conoce a A y C - Router B:
S0 → conectado a Router C a través de la red 162.11.6.0
S1 → conectado a Router A a través de la red 162.11.8.0
E0 → conectado a la red 162.11.7.0
- C conoce solo a B - Router C:
S0 → conectado a la red 162.11.10.0
S1 → conectado a Router B a través de la red 162.11.6.0

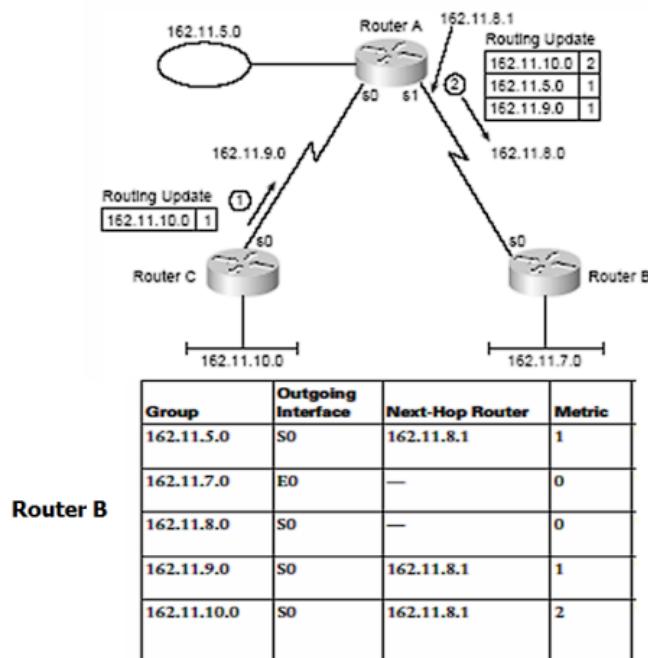


Figura 11: Estado Inicial de Router B.

Después de intercambiar información:

- A aprende que puede llegar a C a través de B.
- C aprende que puede llegar a A a través de B.
- B ya conoce ambos extremos.

Así, cada router va construyendo su visión completa de la red, eligiendo siempre la ruta con menor número de saltos.

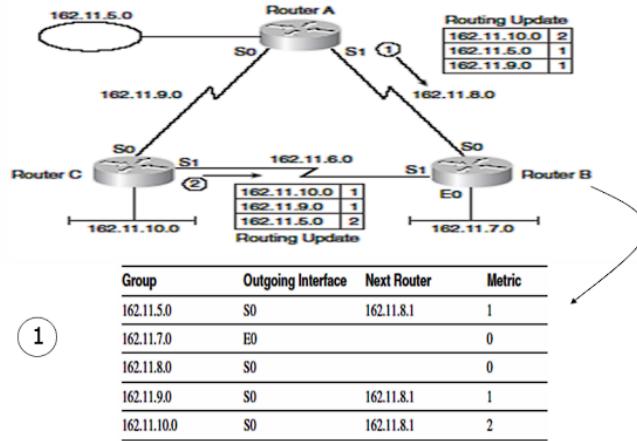


Figura 12: Router B envía actualización a C.

Indica que Router B ha recibido actualizaciones de Router A y está encaminando tráfico a través de él. Por ejemplo, para llegar a la red 162.11.5.0, se dirige a 162.11.8.1 (Router A) con métrica 1.

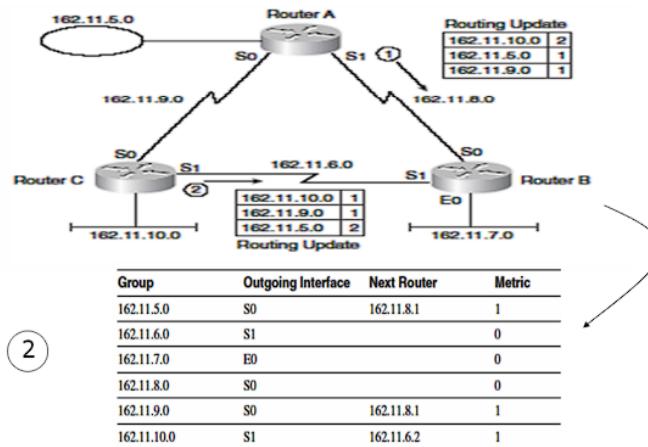


Figura 13: Router C envía actualización a B.

Indica que el Router C también ha aprendido algunas rutas desde Router A. Tiene rutas directas para 162.11.6.0 y 162.11.10.0, que están conectadas directamente.

Cuenta a Infinito
Split Horizon
Mas mejoras

- ✓ Se asocia un timer con cada entrada de la tabla
- ✓ Si no se actualiza la entrada al cabo del temporizador entonces se marca con infinito dicha entrada
- ✓ Normalmente se lo configura 6 veces el intervalo de transmisión
- ✓ Un cambio en la tabla provoca la publicación del mismo, "triggered update"
- ✓ Publicación de tablas cada 90 segundos
- ✓ Hold-down: Si se recibe un incremento en alguna ruta se pasa su métrica a infinito durante el timer de hold-down
- ✓ Si luego se recibe el incremento nuevamente se lo adopta

Figura 14: Enter Caption

2.3.1. Protocolo RIP

2.4. Algoritmo de Link State

- Cada nodo recopila información sobre el estado de sus enlaces directos.
- Difunde esta información a todos los routers del sistema autónomo.
- Construye un mapa completo de la red.
- Utiliza el algoritmo de Dijkstra para calcular caminos óptimos.
- Ofrece mayor precisión y velocidad de convergencia.

2.4.1. Protocolo OSFP

Característica	Vector de Distancia	Estado de Enlace
Info. intercambiada	Tablas con vecinos	Estado de enlaces
Algoritmo base	Bellman-Ford	Dijkstra
Conocimiento de la red	Parcial	Completo
Convergencia	Lenta	Rápida
Complejidad	Baja	Alta
Ejemplos de protocolo	RIP	OSPF

Cuadro 1: Comparación entre los algoritmos de ruteo dinámico.

3. Arquitectura

3.1. Inicios y gestión de Internet

Arpanet

Fue creada en 1969 por la *ARPA* (Advanced Research Projects Agency) del *Department of Defense* (DoD) de los Estados Unidos. Su desarrollo marcó el inicio de las redes de commutación de paquetes y sentó las bases de Internet. Se trataba de una red **no orientada a conexión**, basada en el envío de datagramas. La primera experiencia operativa tuvo lugar en diciembre de 1969, con únicamente cuatro nodos conectados.

El funcionamiento se basaba en *routers* o *IMPs* (Interface Message Processors), los cuales se enlazaban a través de líneas telefónicas de 56 Kbps. Cada IMP se conectaba localmente a un host (computadora principal). El mantenimiento de la subred —formada por los IMPs y las líneas de conexión— fue encargado a la empresa BBN (*Bolt, Beranek & Newman*).

Durante sus primeros años, se intentó adaptar los protocolos iniciales para que funcionaran con redes muy diversas (satélite, radio, etc.), pero se comprobó que no eran adecuados. Para resolver esta limitación, en 1974 Vinton Cerf y Robert Kahn diseñaron los protocolos **TCP/IP**, que posteriormente se convertirían en el estándar para la comunicación en redes.

ARPANET, NSFNET e Internet

La versatilidad del protocolo **TCP/IP** para interconectar tanto redes **LAN** como **WAN**, sumada a su promoción por parte de la ARPA (distribución gratuita junto con el sistema *UNIX BSD 4.2*), provocó un gran crecimiento de ARPANET. Sin embargo, dado que ARPANET estaba financiada por el *Department of Defense* (DoD) de los Estados Unidos, su uso estaba restringido a centros de investigación con proyectos militares.

En 1984, la **NSF** (*National Science Foundation*) creó la red **NSFNET**, abierta a todas las universidades. Esta red se interconectó con ARPANET, permitiendo ampliar su alcance. Con el tiempo, a NSFNET se fueron conectando redes regionales y de otros países, lo que dio lugar a la creación de la **Internet**.

En 1990, ARPANET fue dada de baja y NSFNET pasó a ser gestionada por la empresa **ANS** (*Advanced Networks and Services*). En 1995, ANSNET fue vendida a *America Online*, que en el año 2000 se fusionó con *Time Warner*.

En 1996, un conjunto de universidades estadounidenses puso en marcha **Internet 2**, una nueva red destinada a actividades de investigación y desarrollo (**I+D**) al margen de la Internet comercial. Internet 2 se apoyaba en las infraestructuras *backbone* de **vBNS** (*Very high performance Backbone Network Service*) y **Abilene**.

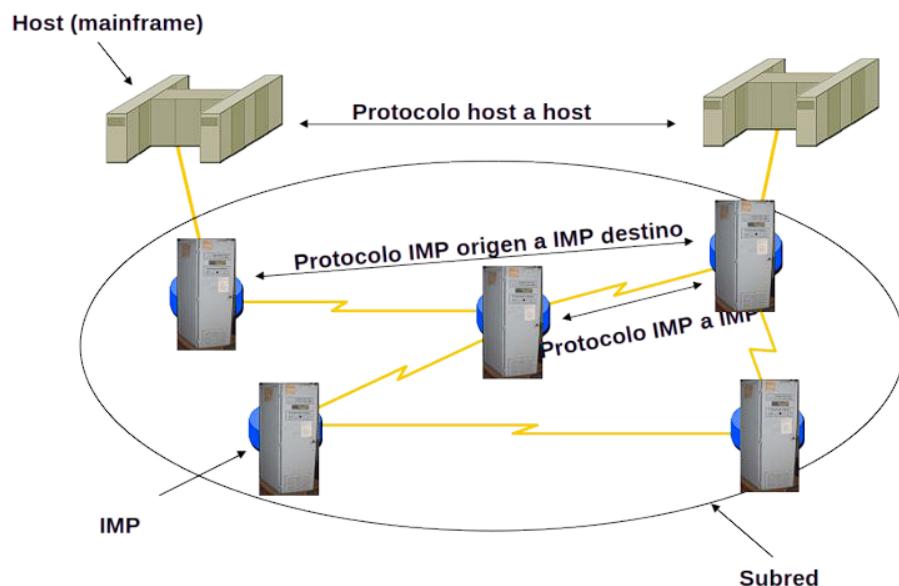


Figura 15: Arquitectura original.

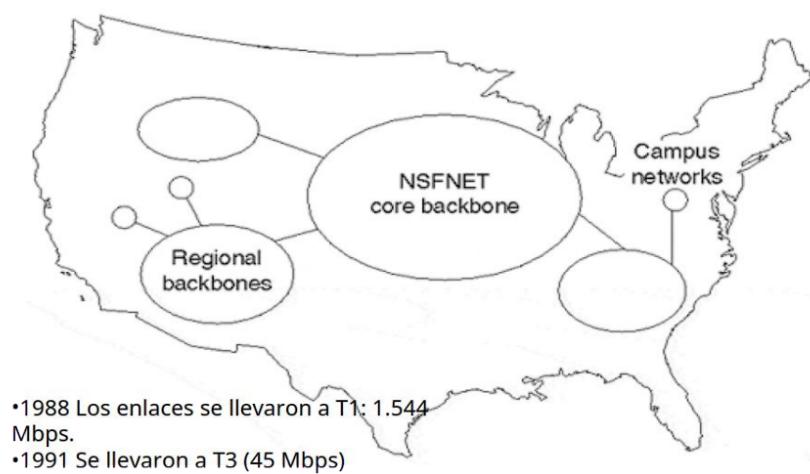


Figura 16: NSFNET-1991.

Definición de Internet (FNC, 1995)

RESOLUTION: El *Federal Networking Council* (FNC) acordó que el siguiente texto refleja su definición oficial del término “**Internet**”:

Internet se refiere al sistema global de información que:

- (I) Está interconectado de manera lógica mediante un espacio de direcciones único a nivel global, basado en el Protocolo de Internet (**IP**) o sus extensiones/derivaciones posteriores.
- (II) Es capaz de soportar comunicaciones utilizando el conjunto de protocolos **TCP/IP** o sus extensiones/derivaciones posteriores, y/o cualquier otro protocolo compatible con IP.
- (III) Proporciona, utiliza o pone a disposición —de forma pública o privada— servicios de alto nivel basados en la infraestructura de comunicaciones y recursos relacionados aquí descritos.

Fecha de aprobación: 24 de octubre de 1995.

ISOC (Internet Society)

En 1991 se creó la **ISOC** (*Internet Society*), una asociación internacional dedicada a la promoción de la tecnología y los servicios de Internet. Cualquier persona física interesada puede asociarse a la ISOC.

La ISOC está gobernada por un **Consejo de Administración** (*Board of Trustees*), cuyos miembros son elegidos por votación entre todos los socios.

El desarrollo técnico de Internet está a cargo del **IAB** (*Internet Architecture Board*), cuyos miembros son nombrados por el Consejo de Administración de la ISOC.

El IAB supervisa el trabajo de dos comités:

- **IRTF** (*Internet Research Task Force*): Se enfoca en estrategias y problemas a largo plazo.
- **IETF** (*Internet Engineering Task Force*): Aborda los problemas técnicos más inmediatos.

El **IESG** (*Internet Engineering Steering Group*) es el responsable de la gestión técnica de las actividades del IETF y de los procesos de estandarización de Internet. Está compuesto por los *Directores de Área* de los grupos de trabajo del IETF.

Enlaces de referencia

- <https://www.isoc.org>
- <https://www.ietf.org>

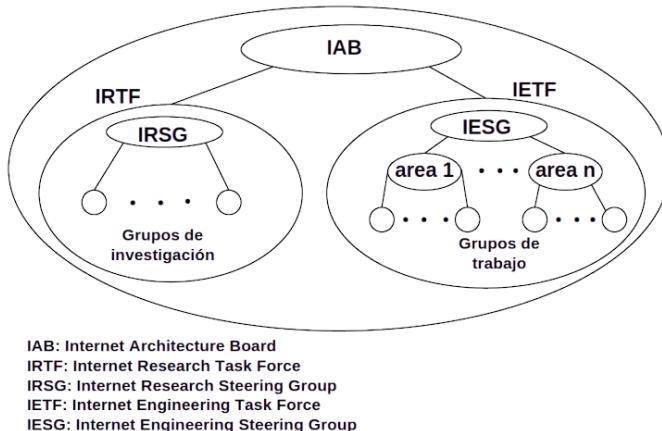


Figura 17: Organización del trabajo técnico.

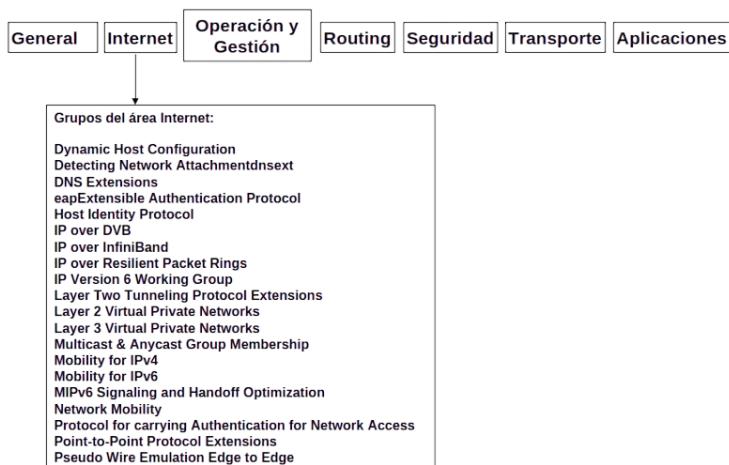


Figura 18: Áreas del IETF.



Figura 19: IANA (Internet Assigned Numbers Authority).

3.2. Interconexión

Un **Sistema Autónomo** es un conjunto de routers, equipos y sistemas conectados a Internet bajo un control administrativo común.

BGP (Border Gateway Protocol)

- Redistribuye la información de subredes en toda la extensión de Internet.
 - Del orden de 300.000 entradas.
- Funciona entre Sistemas Autónomos.
- Permite implementar **acuerdos de Peering** entre SAs:
 - **Atributos usados:** local-preference, multi exit discriminators (MEDs), AS-Path.
 - **Peering policy:** plantilla ajustada a medida del acuerdo o contrato.
- Permite implementar otros acuerdos privados:
 - Con clientes con requerimientos especiales.
 - Con clientes u otros ISPs con requerimientos de QoS.
 - Atributos usados: Community.
- Permite redistribuir información de subredes IPv4 con IPv6.

Peering vs Transit

Peering es la interconexión voluntaria de redes administrativamente independientes para intercambiar tráfico de sus usuarios finales.

- Normalmente sin intercambio monetario.
- Beneficios: reducción de costos, mejor rendimiento, mayor control de rutas, redundancia y colaboración técnica.

Transit en este caso, una red paga a otra para acceder al resto de Internet. Permite conectividad global sin necesidad de acuerdos con cada red.

Principio de alcanzabilidad extremo-a-extremo

Una red está conectada a Internet si compra tránsito o tiene *peering* con todas las redes que no compran tránsito. Las redes sin tránsito forman la **Default Free Zone (DFZ)**.

Motivaciones para el Peering

1. Beneficio mutuo y reducción de costos.
2. Redundancia ante fallos.

3. Mayor capacidad para tráfico masivo.
4. Control de enrutamiento.
5. Mejor rendimiento evitando cuellos de botella.
6. Mejor imagen de red (posible ascenso de *tier*).
7. Facilidad de cooperación con peers.

Clasificación de redes: Tier 1, Tier 2 y Tier 3

No existe una autoridad formal que defina esta clasificación; es un consenso de la industria basado en la conectividad y dependencia entre redes.

- **Tier 1:** Red que puede alcanzar cualquier otra red de Internet **sin pagar** tránsito (IP transit) ni peering. Mantiene acuerdos de *peering* gratuito (*settlement-free peering*) con todos los demás Tier 1. Forma parte de la **Default Free Zone (DFZ)**. *Ejemplos:* Lumen (Level 3), AT&T, NTT.
- **Tier 2:** Red que realiza *peering* gratuito con algunas redes, pero compra tránsito a otras para alcanzar todo Internet. *Ejemplos:* ISPs regionales grandes como Telefónica o Cogent.
- **Tier 3:** Red que compra todo el tránsito a Tier 1 o Tier 2 para poder conectarse a Internet. *Ejemplos:* Proveedores locales de Internet.

Peering público	Peering privado
Shared fabric. Múltiples carriers acuden al sitio de interconexión participando con uno o más interfaces físicas	Interconexión directa entre sólo dos redes, a nivel de Layer 1 o 2, con una capacidad dedicada.
Antes llamados Network Access Points (NAPs) y hoy conocidos como Exchange Points o Internet exchanges ("IXP").	Antiguamente ocurrían sobre un circuito dedicado, hoy mayormente en sitios de colocación privados o neutrales donde se hacen las cros-conexiones directas a un menor costo

Figura 20: Peering público vs privado.

3.3. Tipos de redes

Según su finalidad las redes se clasifican en:

- **Redes de servicios privados** → corporativas, bancarias, industriales.
- **Redes de contenidos (CDN)** → para distribución de video, música, actualizaciones de software.

- **Redes de investigación y educación** → universidades, centros de I+D.
- **Redes gubernamentales** → para administración pública, defensa, seguridad.
- **Redes comunitarias** → creadas por usuarios para compartir acceso a Internet o recursos (ej. guifi.net).
- **Redes críticas** → infraestructuras como energía, transporte, salud.
- **Redes financieras** → especializadas en transacciones y mercados (ej. SWIFT).

Las Redes de servicios privados, de contenido y de investigación son los mas relevantes para entender cómo se usa la red en el mundo real hoy, sobre todo en áreas de contenido, seguridad y colaboración científica.

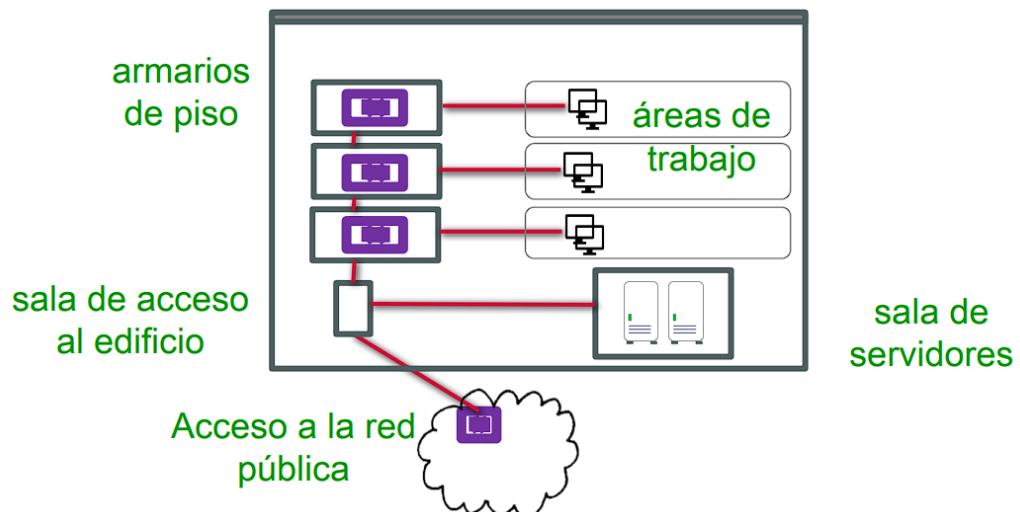
Redes de servicios privados

La Calidad de Servicio (QoS) se conforma de:

- **Requerimientos de seguridad:** Proteger los datos y los servicios frente a accesos no autorizados, ataques, malware, etc.
- **Requerimientos de disponibilidad:** Asegurar que el servicio esté accesible la mayor parte del tiempo (mínimos tiempos de caída).
- **Requerimientos de retardo:** Minimizar el tiempo que tarda un paquete en ir del origen al destino, crucial para voz, video y juegos online.
- **Requerimientos de capacidad:** Garantizar suficiente ancho de banda para el tráfico esperado.

Para cumplir con los requerimientos de calidad de servicio, surgen otros dispositivos y tecnologías tales como:

- **FW (Firewalls):** Control y filtrado del tráfico de red.
- **Inspectores de tráfico:** Analizan los paquetes en tiempo real.
- **DMZs (Demilitarized Zones):** Segmentan la red para mayor seguridad.
- **Balanceadores de tráfico:** Distribuyen la carga entre múltiples servidores.
- **Aceleradores WAN:** Optimización del rendimiento en redes de área amplia.



Cableado estructurado: vertical vs. horizontal

Figura 21: Redes enterprise con Datacenter en sitio.

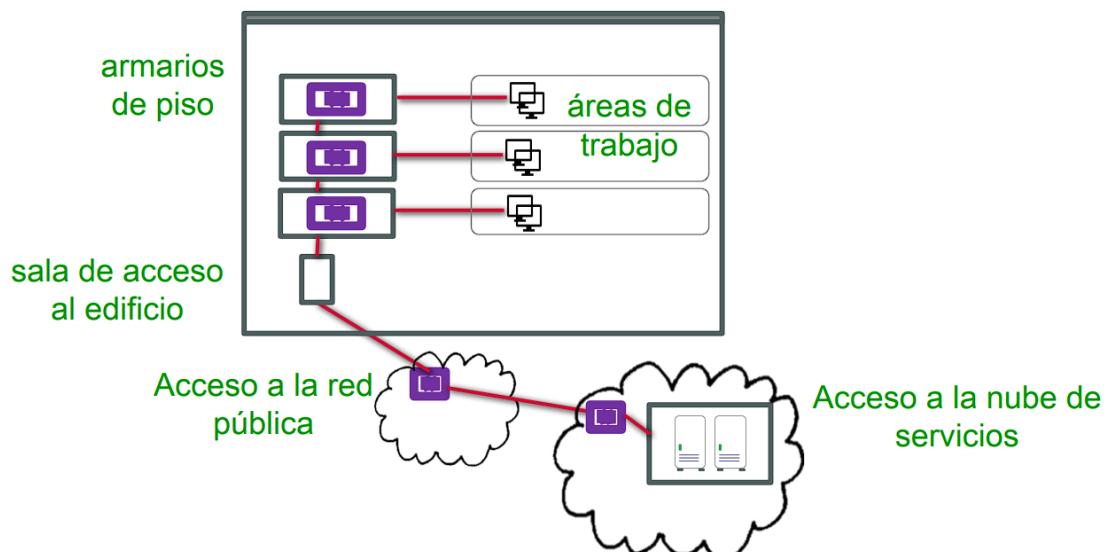


Figura 22: Redes enterprise con Datacenter en la nube (caso 1).

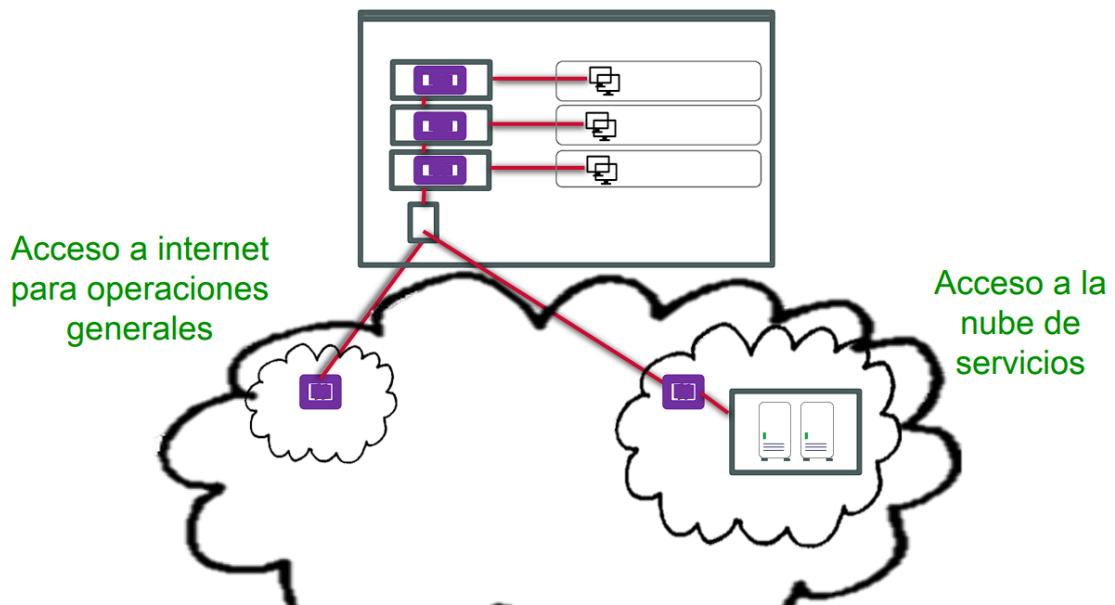


Figura 23: Redes enterprise con Datacenter en la nube (caso 2).

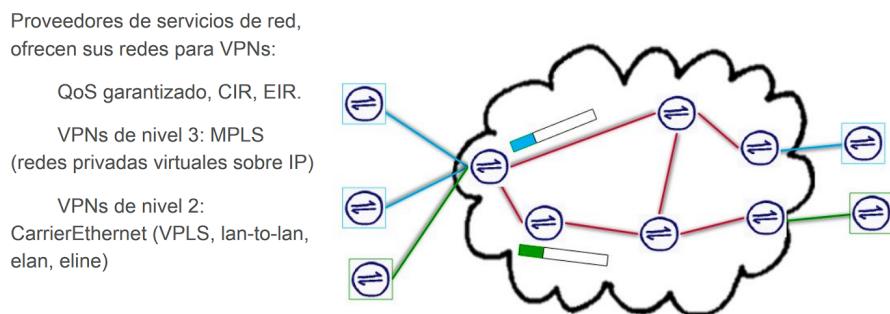


Figura 24: Redes de servicio privado.

Redes de contenidos

Las redes de contenidos pueden ser clasificadas en diferentes tipos según su acceso y características:

Las **redes privadas** están destinadas a la distribución de contenido como descargas de software, libros, y otros archivos, generalmente restringidos a usuarios autorizados. Por otro lado, las **redes públicas** pueden ser utilizadas para eventos específicos o mantenerse de forma permanente, ofreciendo contenido accesible para un público amplio.

Estas redes también se organizan por **regiones**, permitiendo que el **contenido varíe según la ubicación geográfica** de los usuarios, optimizando así la distribución y adecuación del material ofrecido.

Un concepto importante en estas redes es el de **open-caching**, que permite almacenar en caché contenidos en ubicaciones abiertas para mejorar la eficiencia de entrega.

Además, manejan **fechas de validez** para controlar el tiempo durante el cual un contenido permanece disponible o actualizado.

Se establecen también diferentes tipos de canales, distinguiendo entre el **canal de control**, que se encarga de la gestión y señalización de la red, y el **canal de datos**, por donde realmente se transmite el contenido, ambos con requerimientos específicos, incluyendo en algunos casos la necesidad de soporte para **requerimientos en tiempo real (real-time)** para aplicaciones sensibles a la latencia.

Redes de investigación

Las redes de investigación se caracterizan por contar con una **infraestructura de alta capacidad**, diseñada para soportar grandes volúmenes de datos y ofrecer conexiones rápidas y confiables. Estas redes suelen tener **gestión propia**, es decir, son administradas por las instituciones académicas o consorcios que las utilizan, lo que permite un control directo sobre su funcionamiento y seguridad. Además, cuentan con **puntos de interconexión** estratégicos con Internet y otras redes académicas, facilitando el intercambio eficiente de información entre centros de investigación y universidades. Otro aspecto importante es que estas redes pueden ofrecer **contenidos exclusivos** relacionados con proyectos científicos y académicos, accesibles únicamente para sus usuarios autorizados, lo que fomenta la colaboración y el avance en la investigación.

Futuro: Solar System Internet

El concepto de **Solar System Internet** plantea una red de comunicaciones que abarque no solo la Tierra, sino también otros cuerpos del sistema solar. Este avance presenta varios desafíos y necesidades particulares.

Uno de los principales retos es si los desafíos serán **iguales o diferentes** a los que enfrentan las redes terrestres actuales, considerando las condiciones espaciales. Las **necesidades** específicas incluyen la gestión de la **distancia** entre dispositivos, ya que las enormes separaciones afectan la transmisión de datos.

La comunicación requiere **línea de vista** y sistemas de **tracking** para mantener enlaces estables entre satélites y estaciones espaciales. Además, el **peso y volumen** de los equipos que se despachan al espacio son limitados, por lo que es crucial optimizar el diseño y uso de materiales.

La **latencia**, incluso a la velocidad de la luz, se convierte en un factor importante debido a las grandes distancias, afectando la rapidez de la comunicación.

Otro desafío es el **(auto)mantenimiento** del equipamiento, incluyendo satélites, routers y software, que debe ser capaz de operar de forma autónoma en condiciones extremas.

Finalmente, la **defensa frente a ataques** ciberneticos o físicos es fundamental para proteger esta infraestructura crítica en un entorno espacial hostil.

DTN

Delay / disruption-tolerant network

RFC 4838

Interplanetary Overlay Network (ION)

(NASA's DTN) Bundle protocol

RFC 5050

Lectura complementaria: An Approach
to Interplanetary Internet.pdf

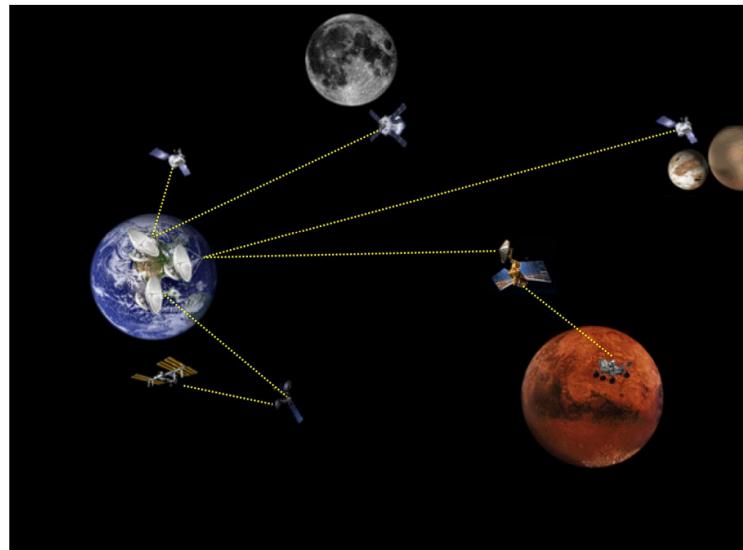


Figura 25: Comunicación en redes espaciales.

4. Redes Definidas por Software

Plano de control y plano de datos: Redes tradicionales

La función principal del nivel de red, en una red IP, es transportar datagramas de un host origen a un host destino. El nivel de red se puede dividir en dos partes que interactúan entre sí:

- El plano de control.
- El plano de datos.

Routing: Determina el camino que van a realizar los datagramas desde un punto de la red a otro. Implementado con software para reducir costos.

Forwarding: Acción local del router de transferir un datagrama de una interfaz de entrada a una de salida. Comúnmente implementado con hardware para reducir demoras.

El algoritmo de los protocolos de ruteo determinan el contenido de las tablas de Forwarding de los routers.

4.1. Arquitectura de un Router

La arquitectura de un router se compone de tres elementos principales: el procesador de ruteo, los puertos (de entrada y salida) y el *switch fabric*. Cada uno cumple funciones específicas dentro del encaminamiento de paquetes.

Procesador de ruteo (*Routing Processor*)

Es el encargado de las funciones del plano de control y de la administración general del router. Sus principales tareas incluyen:

- Ejecutar protocolos de ruteo (por ejemplo, OSPF, BGP, RIP).
- Mantener las tablas de ruteo y la información del estado de los enlaces.
- Calcular la tabla de *forwarding* utilizada en el plano de datos.
- Realizar funciones de gestión y supervisión de la red.

Puertos

Los puertos del router se dividen en:

- **Puertos de entrada:** Encargados de las funciones físicas (nivel físico), de enlace (nivel de enlace de datos) y de la búsqueda en la tabla de *forwarding* para determinar el puerto de salida correspondiente.
- **Puertos de salida:** Reciben las tramas desde el *switch fabric*, las almacenan temporalmente y las transmiten hacia el enlace de salida, realizando las funciones necesarias de los niveles físico y de enlace.

Switch Fabric

Es el sistema interno que conecta los puertos de entrada con los puertos de salida, permitiendo que los paquetes fluyan a través del router. Su diseño y capacidad determinan el rendimiento global del equipo.

4.2. SDN

En su concepción original, una *Software-Defined Network* (SDN) se caracteriza por dos principios fundamentales:

- **Separación física del plano de control y el plano de datos:** el plano de control se encarga de tomar las decisiones sobre el encaminamiento y gestión del tráfico, mientras que el plano de datos se limita a reenviar los paquetes según dichas decisiones.
- **Control centralizado:** un único plano de control coordina todos los dispositivos de *forwarding* de la red, lo que simplifica la administración y la implementación de políticas.

Características principales de la arquitectura SDN

Las principales características de una arquitectura de Redes Definidas por Software (SDN) son:

- **Forwarding basado en flujos:** el encaminamiento y tratamiento de los paquetes se realiza siguiendo reglas definidas en tablas de flujos.
- **Elementos de forwarding simplificados:** los dispositivos se limitan a aplicar las reglas recibidas sin ejecutar algoritmos complejos de ruteo.
- **Separación del plano de control y el plano de datos:** el plano de control se ejecuta de forma independiente al hardware encargado del reenvío de paquetes.
- **Funciones de control centralizadas:** toda la lógica y políticas de la red se ejecutan en servidores o controladores externos a los switches.
- **Control centralizado:** un único controlador SDN gestiona y supervisa el comportamiento de toda la red.
- **Red programable mediante APIs:** la administración y automatización de la red se realiza a través de interfaces de programación de aplicaciones.

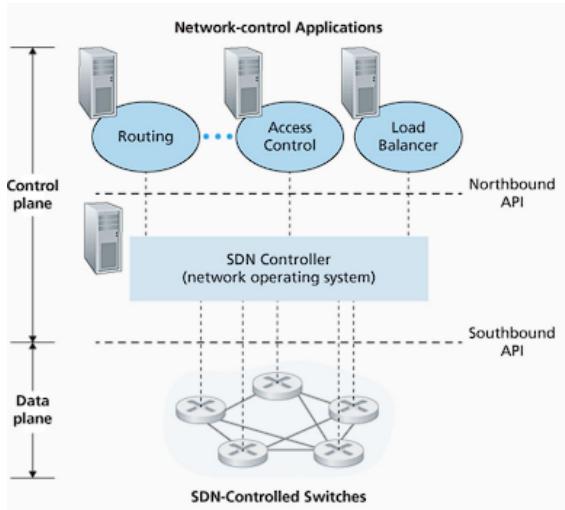


Figura 26: Arquitectura SDN.

Historia

Tradicionalmente, las funciones de control y reenvío se encontraban integradas en el hardware de los dispositivos de red (por ejemplo, routers y switches). Con el tiempo, los avances tecnológicos permitieron realizar funciones más complejas directamente en el hardware, mejorando la velocidad y reduciendo costos.

En 2008, un trabajo académico [?] presentó el protocolo **OpenFlow**, que se convirtió en el primer estándar ampliamente adoptado para la comunicación entre el plano de control y el plano de datos en arquitecturas SDN.

A partir de allí surgieron los llamados *bare-metal switches* o *whitebox switches*, dispositivos de red con hardware genérico que pueden ser controlados por software externo, independientemente del fabricante.

Con el tiempo, SDN ha evolucionado más allá de su definición inicial, incorporando también la capacidad de contar con un plano de datos **programable**, permitiendo modificar la lógica de reenvío mediante lenguajes como *P4*.

Arquitectura

En una arquitectura SDN típica, el control de la red se centraliza en un **controlador SDN**, que se encuentra físicamente separado de los switches y routers. Este controlador:

- Genera y mantiene las *tablas de flujo* (*flow tables*) que definen cómo deben procesarse los paquetes.
- Distribuye estas tablas a los dispositivos de *forwarding* mediante protocolos como **OpenFlow** o el más reciente **P4**.
- Supervisa el estado de la red y ajusta dinámicamente las reglas de reenvío según sea necesario.

Los **switches SDN** almacenan estas tablas de flujo y actúan únicamente según las reglas establecidas por el controlador, lo que permite una administración centralizada, flexible y programable de toda la infraestructura de red.

Interfaces Norte y Sur

Interfaz Norte: Es una API que permite tener acceso a información de la topología de bajo nivel red (dispositivos, enlaces y hosts), y proporcionan una variedad de abstracciones para afectar el estado de la red.

Interfaz Sur: Es una API a través de la cual el núcleo del controlador interactúa con el entorno de red. Ejemplos de esta interacción son por ejemplo la obtención de estadísticas y la modificación del comportamiento de los switches. Los dos protocolos que pueden realizar esta función son OpenFlow y P4.

Forwarding Generalizado: *Match-Plus-Action*

Los routers y switches utilizan tablas de *forwarding* para decidir qué hacer con cada paquete entrante. Una abstracción ampliamente utilizada para describir este proceso es el modelo *match-plus-action*, que consiste en dos pasos fundamentales:

1. **Match (coincidencia)**: Comparar ciertos bits o campos específicos del encabezado del paquete con las entradas de la tabla de *forwarding*.
2. **Action (acción)**: Ejecutar una o más acciones según la coincidencia encontrada.

Forwarding basado en destino

En el caso más tradicional, conocido como *destination-based forwarding*, el reenvío se basa únicamente en la dirección IP de destino. La acción típica consiste en enviar el paquete al siguiente salto (*next hop*) correspondiente en la ruta hacia su destino final.

Forwarding generalizado

En el *forwarding generalizado*, la decisión no se limita a la dirección de destino, sino que puede basarse en múltiples campos del encabezado, como:

- Dirección IP de origen.
- Números de puerto TCP/UDP.
- Tipo de protocolo.
- Etiquetas VLAN, MPLS, etc.

Además, las acciones posibles son mucho más amplias, incluyendo:

- Reenviar el paquete a un puerto específico.
- Descartar el paquete.

- Copiarlo a otro puerto para monitoreo o análisis.
- Modificar ciertos campos del encabezado.
- Registrar información para *logging*.

Este enfoque permite mayor flexibilidad y control en la gestión del tráfico, siendo un concepto clave en tecnologías como SDN (*Software-Defined Networking*) y en la implementación de políticas avanzadas de red.

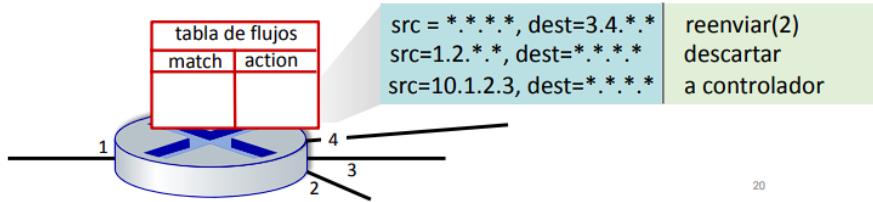


Figura 27: Tabla de flujos.

4.3. Protocolo OpenFlow

OpenFlow es un protocolo abierto que permite implementar de manera práctica la separación entre el plano de control y el plano de datos en una red. Fue pionero en la aplicación de los conceptos de *forwarding* generalizado y en el desarrollo de las Redes Definidas por Software (SDN). La primera versión oficial de OpenFlow se liberó en febrero de 2011, marcando un hito en la evolución de las arquitecturas de red programables.

En una red basada en SDN, el **controlador** actúa como el cerebro del sistema, comunicándose con los dispositivos de *forwarding* (como routers y switches) para tomar decisiones sobre el ruteo y la gestión del tráfico. A través de OpenFlow, el controlador puede instalar y configurar las tablas de flujos en estos dispositivos, definiendo de manera precisa cómo deben procesarse los paquetes.

El protocolo OpenFlow opera sobre TCP, utilizando por defecto el puerto 6653, en el que el controlador permanece a la escucha de las conexiones provenientes de los dispositivos de red. Este canal permite intercambiar mensajes de control, actualizar reglas de *forwarding* y recopilar estadísticas, garantizando así una gestión centralizada y flexible de la red.

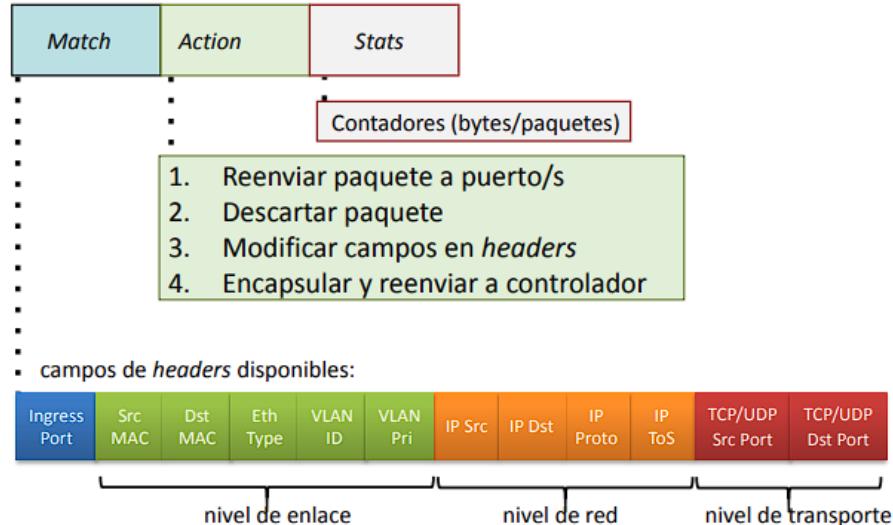


Figura 28: Entradas de la tabla de flujos.

Destination-based forwarding

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	51.6.0.8	*	*	*	port6

Los datagramas IP destinados a la dirección 51.6.0.8 deben ser reenviados al puerto de salida 6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	*	drop

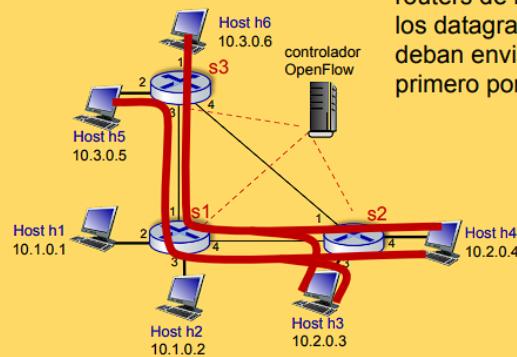
Descartar todos los datagramas destinados al puerto TCP 22 (ssh)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	128.119.1.1	*	*	*	*	drop

Descartar todos los datagramas enviados por el host 128.119.1.1

Figura 29: Ejemplos de OpenFlow.

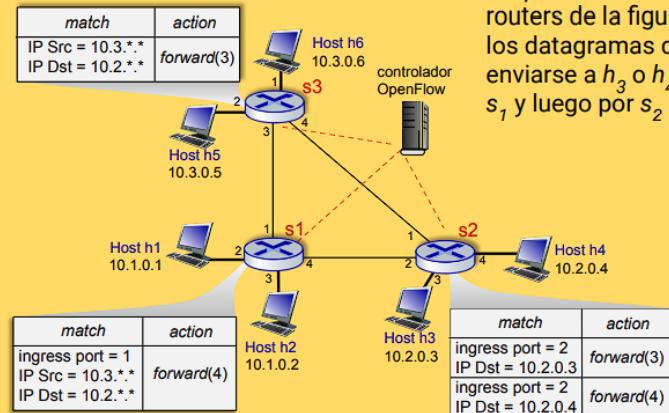
SDN - Ejercicio



Proponer tablas de flujos para los routers de la figura de forma tal que los datagramas de h_5 y h_6 que deban enviarse a h_3 o h_4 pasen primero por s_1 y luego por s_2

Figura 30: Ejercicio SDN.

SDN - Ejercicio



Proponer tablas de flujos para los routers de la figura de forma tal que los datagramas de h_5 y h_6 que deban enviarse a h_3 o h_4 pasen primero por s_1 y luego por s_2

Figura 31: Resolución.

Switch OpenFlow

Un **Switch OpenFlow** se compone de los siguientes elementos:

- Una o más **tablas de flujo**, responsables de realizar la búsqueda (*packet lookup*) y el reenvío de paquetes (*forwarding*).
- Uno o más **canales de comunicación** con el controlador SDN, utilizando el protocolo OpenFlow. A través de estos canales, el controlador puede agregar, actualizar o eliminar entradas en la tabla de flujos, de forma tanto reactiva como proactiva.

Un **flujo** describe un conjunto de paquetes que se transfieren desde un dispositivo de la red hacia otro, cumpliendo con ciertos criterios definidos por los campos de los encabezados de los paquetes.

4.4. P4

Contexto previo a P4, tradicionalmente, los chips de switches de alto desempeño soportaban un conjunto fijo de funciones, implementando únicamente los protocolos definidos por IEEE y la IETF.

En la época en que surgió OpenFlow, todos los switches ofrecían básicamente las mismas funciones: Ethernet, IPv4, VLANs, etc. OpenFlow asume que los switches poseen un comportamiento fijo, descrito en la hoja de datos del ASIC (*Application-Specific Integrated Circuit*).

El protocolo OpenFlow no controla el comportamiento interno del switch, sino que permite utilizar un conjunto predeterminado de funciones ya implementadas en el hardware.

Limitaciones y avances

En sus inicios, los chips de switches programables procesaban tramas entre 10 y 100 veces más lento que un ASIC de funciones fijas. Con la aparición de los chips **PISA** (*Protocol Independent Switch Architecture*) esta limitación desapareció, ofreciendo un rendimiento comparable al de los ASIC tradicionales.

La incorporación de un nuevo protocolo en un ASIC de funciones fijas podía demorar hasta 4 años. Ejemplo actual de avance: *Inband Network Telemetry* (INT).

Nacimiento de P4.

Surge la necesidad de definir de forma precisa cómo debe procesarse un paquete dentro del switch. En 2013, un consorcio integrado por Google, Intel, Microsoft, Stanford, Princeton y Barefoot desarrolló el lenguaje **P4**.

P4 permite especificar al switch qué debe hacer y cómo procesar los paquetes, en lugar de limitarse al conjunto fijo de funciones soportadas por el hardware. Con P4 es posible definir qué encabezados reconocerá un switch y qué acciones debe ejecutar sobre ellos.

P4Runtime

La **API P4Runtime** es una especificación del plano de control para gestionar elementos del plano de datos de un dispositivo definido o descrito mediante un programa P4.

4.5. Open Compute Project (OCP)

OCP es una comunidad colaborativa centrada en el rediseño de la tecnología de hardware para soportar de manera eficiente las demandas de infraestructura informática.

Certifica hardware sobre el que se ejecuta software SDN.

Podemos hacer una analogía con una PC construida a partir de componentes listos para usar. Así como podemos comprar servidores Dell y HP, también podemos comprar switches preconstruidos (compatible con OCP) de proveedores de switches sin sistema operativo como EdgeCore, Delta y otros.

Este hardware equivalente al software de código abierto y hace posible que cualquiera pueda construir un switch de alto rendimiento, análogo a una PC construida en casa.

En Facebook o Google, que tienen miles de switches, sería carísimo y poco flexible usar routers propietarios. Con OCP, compran hardware abierto y les instalan su propio software de red hecho a medida. Usar hardware OCP en vez de un router tradicional tiene sentido sobre todo en centros de datos grandes, empresas de telecomunicaciones o redes de proveedores de nube, debido a que:

- **Costo más bajo a gran escala:** Ideal para empresas con miles de switches, evitando el alto costo de routers propietarios.
- **Independencia del fabricante:** Permite elegir hardware y software sin quedar atado a un proveedor específico.
- **Flexibilidad y personalización:** Se puede instalar software de red propio y adaptarlo a las necesidades.
- **Optimización para cargas específicas:** Configurable para maximizar el rendimiento en tareas concretas.
- **Integración con SDN y virtualización:** Compatible con arquitecturas de red definidas por software y entornos virtualizados.

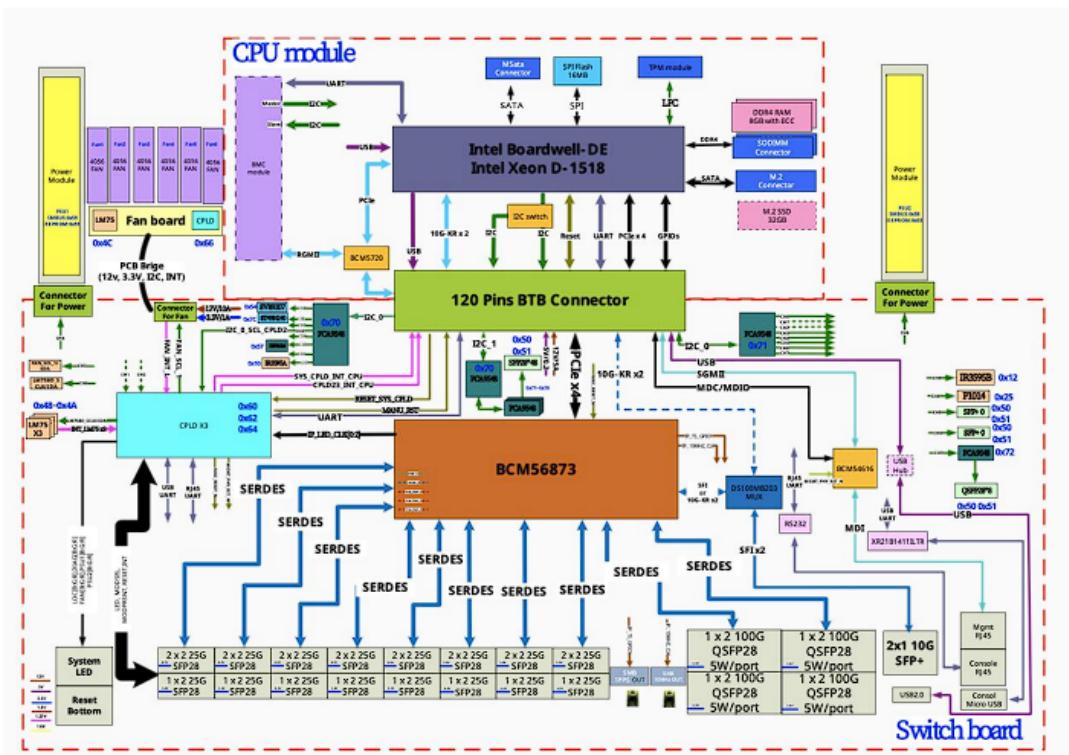


Figura 32: Esquema de un Open Compute Project (OCP).

5. Nivel Físico

Electricidad

- **Velocidad:** Las señales eléctricas se propagan a velocidades menores que la velocidad de la luz (c), dependiendo de las dimensiones físicas del sistema.
- **Datos digitales (ceros y unos):** En una computadora, los bits se representan mediante niveles de tensión (voltajes diferentes para 0 y 1).
- **Tensión (Voltaje):** Es la energía que una carga eléctrica gana o pierde al moverse a través de un circuito.
- **Píxeles del monitor:** Cada píxel emite fotones utilizando la energía proporcionada por las cargas eléctricas.
- **Potencia eléctrica:** Es la energía requerida para mover las cargas eléctricas y para que los píxeles emitan luz (fotones).

La velocidad a la que se propagan las señales eléctricas es menor que la velocidad de la luz (c), y depende de las dimensiones físicas del sistema. Los *ceros* y *unos* de la computadora se representan mediante niveles de tensión, es decir, diferentes voltajes.

La tensión es la energía que una carga eléctrica gana o pierde al moverse a través de un circuito. Consecuentemente, los píxeles de un monitor funcionan emitiendo fotones utilizando la energía que proviene de esas cargas eléctricas. La potencia eléctrica es la energía necesaria para mover esas cargas y para que los píxeles puedan emitir luz (fotones).

Ondas electromagnéticas

La velocidad de propagación de las señales está relacionada con la velocidad de la luz (c) y las dimensiones del sistema. Es importante considerar el retardo en la propagación, que afecta el desempeño de las comunicaciones.

La frecuencia (f) es el parámetro que podemos controlar, mientras que la velocidad del medio (v) es una característica inherente que no podemos modificar. La longitud de onda (λ) se calcula como:

$$\lambda = \frac{v}{f} \quad (1)$$

En el caso de *ondas en espacio libre*, la energía irradiada disminuye con el cuadrado de la distancia, lo que significa que la señal se atenúa rápidamente al alejarse del emisor. Por otro lado, las *ondas guiadas*, como en cables o fibras ópticas, presentan una mínima pérdida de energía durante su propagación.

5.1. Medio físicos

Medio Óptico

La luz es una onda electromagnética; sin embargo, su interacción con la materia requiere un análisis basado en la física cuántica. Esto se debe a que la luz está compuesta por partículas llamadas *fotones*, que son paquetes discretos de energía emitidos o absorbidos por los electrones de un átomo.

Los fotones pueden ser emitidos por dispositivos como LEDs o láseres, y se detectan gracias al efecto fotoeléctrico, que convierte la energía luminosa en señales eléctricas. La intensidad de la luz está determinada por la cantidad de fotones presentes, mientras que la energía de cada fotón es inversamente proporcional a la longitud de onda, es decir, $E \propto \frac{1}{\lambda}$.

Los fotones viajan a la velocidad de la luz (c), y los colores que percibimos están relacionados con la longitud de onda (λ) y la frecuencia (f) de la luz.

	Frequency Range	Attenuation	Delay	Repeater Spacing
Twisted pair (with loading)	0 to 3.5 kHz	0.2 dB/km @ 1 kHz	50 µs/km	2 km
Twisted pairs (multi-pair cables)	0 to 1 MHz	0.7 dB/km @ 1 kHz	5 µs/km	2 km
Coaxial cable	0 to 500 MHz	7 dB/km @ 10 MHz	4 µs/km	1 to 9 km
Optical fiber	186 to 370 THz	0.2 to 0.5 dB/km	5 µs/km	40 km



Figura 33: Medios guiados.

- Medios no guiados
- Transmisión y recepción con una antena

Antena Direccional

- Haz enfocado
- Se requiere alineación



Antena Omnidireccional

- La señal se esparce en todas las direcciones
- Pueden recibirla muchas antenas



30MHz to 1GHz

- omnidireccional
- broadcast radio

2 GHz to 40 GHz

- microondas
- muy direccional
- punto a punto
- satélites

3×10^{11} to 2×10^{14}

- infrarrojo
- alcance local

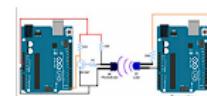


Figura 34: Medios inalámbricos.

5.2. Modulación

Un sistema de comunicación digital completo incluye:

- Procesos en el transmisor para preparar y enviar la señal.
- El canal de comunicación, donde la señal puede sufrir distorsiones.
- Procesos en el receptor para recuperar la información original de manera eficiente y segura.

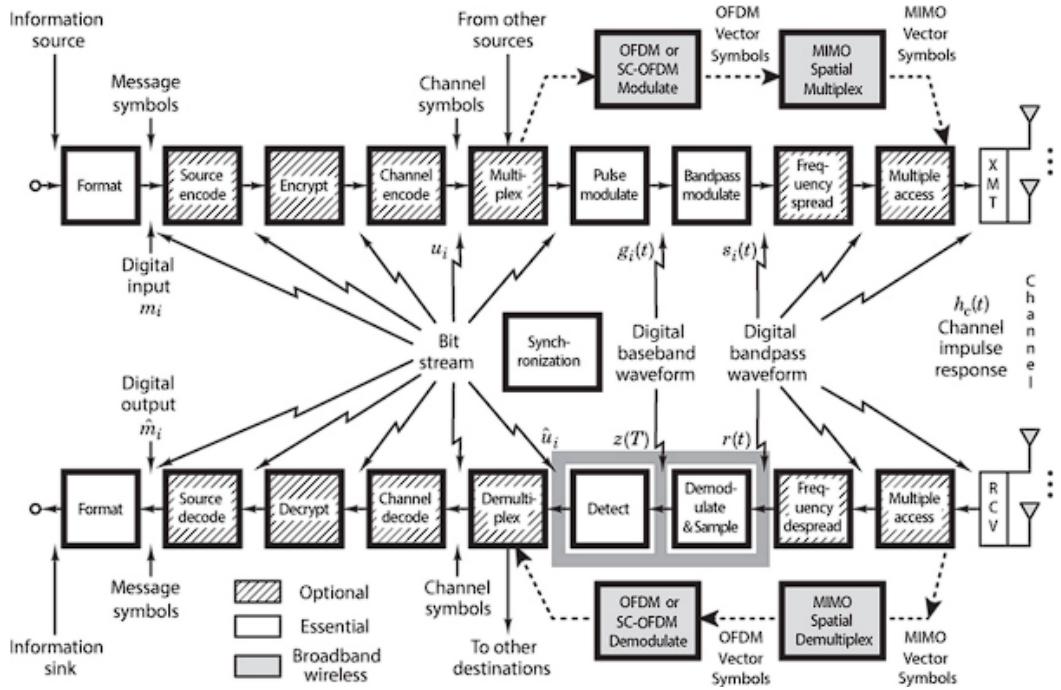


Figura 35: Esquema de un sistema de comunicaciones.

Banda Base

En un sistema de comunicación digital, la *banda base* corresponde a la señal original antes de ser modulada para su transmisión por el canal. El *Bit Stream* representa, de forma conceptual, la secuencia de “ceros” y “unos” que contienen la información a transmitir. Esta secuencia se transforma en una **forma de onda digital de banda base**, normalmente en forma de tensión eléctrica, que codifica dichos bits para su posterior procesamiento o modulación.

Serie de Fourier

Toda señal periódica puede descomponerse en una suma infinita de términos (armónicos) con frecuencias y amplitudes variables, lo que se conoce como *serie de Fourier*.

Transformada de Fourier

Herramienta matemática que permite descomponer cualquier señal, periódica o no, en sus componentes de frecuencia. Es esencial para el análisis de señales en comunicaciones.

Modulación

La *modulación* consiste en modificar uno o varios parámetros de una señal portadora (generalmente una onda senoidal) para transportar la información.

- Señal portadora (carrier): onda que transporta la información.
- Señal moduladora: señal de entrada que contiene la información.
- Señal modulada: resultado de aplicar la modulación.

En **AM** (Amplitude Modulation) se varía la amplitud de la portadora. En **FM** (Frequency Modulation) se varía la frecuencia de la portadora.

La señal portadora por sí sola no tiene información útil más que su forma y frecuencia, pero una vez que se le aplica la moduladora, esa plantilla se moldea para “llevar” el mensaje.

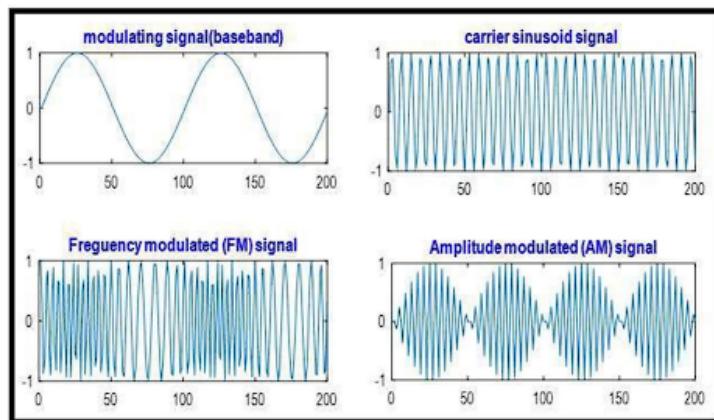
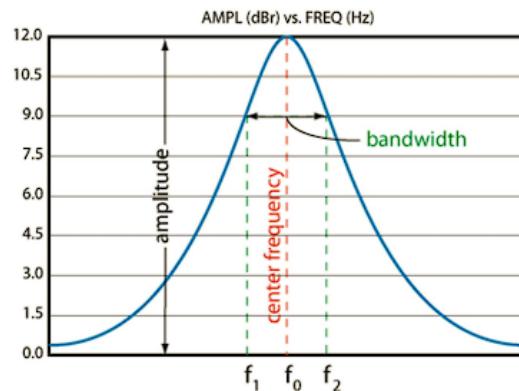


Figura 36: Ejemplo de modulación AM vs FM.



f_0 = frecuencia central

f_1, f_2 = frecuencias de corte inferior y superior (3dB menos que la amplitud máxima)

Ancho de banda (bandwidth) $BW=f_2-f_1$

Figura 37: Ancho de banda de una señal modulada.

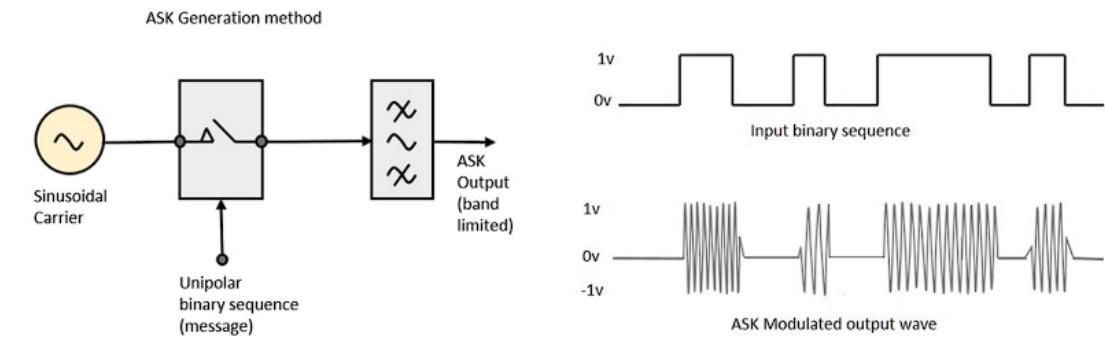


Figura 38: Modulación digital de la amplitud.

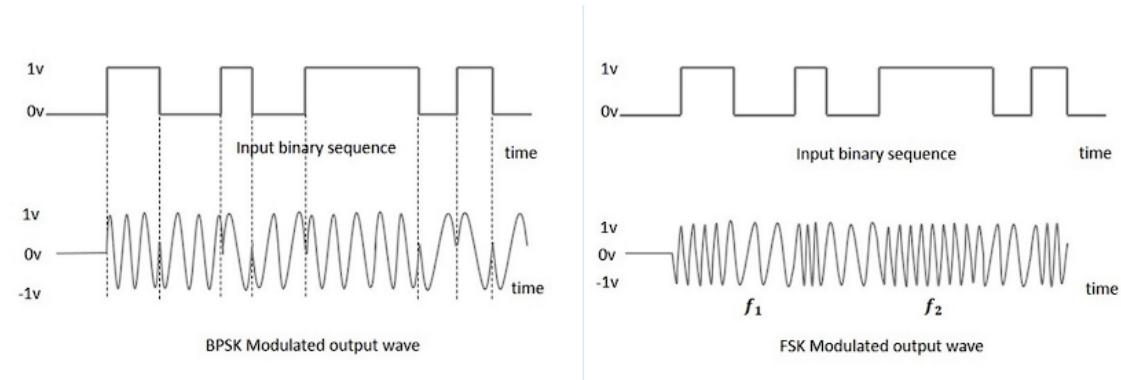


Figura 39: Modulación de la fase y la frecuencia.

5.3. Multiplexación

Multiplexar consiste en compartir un mismo enlace de comunicación entre varias señales o usuarios, cuando este enlace no está siendo utilizado en toda su capacidad, con el objetivo de aumentar la eficiencia.

Este método se utiliza especialmente en troncales (canales que transportan gran volumen de datos) y en enlaces de larga distancia, donde es costoso o impráctico instalar canales separados para cada transmisión.

Las técnicas de multiplexación se pueden aplicar sobre distintos medios físicos, como cable coaxial, enlaces de microondas o fibra óptica, permitiendo que varias señales viajen de forma simultánea sin interferirse. Básicamente, multiplexar es para optimizar la transmisión, y demultiplexar es para restaurar cada flujo a su dueño original.

Multiplexación por División de Frecuencias (FDM)

En este tipo de multiplexación, cada señal se modula utilizando una portadora con una frecuencia distinta. Es fundamental que las frecuencias centrales de las portadoras estén suficientemente separadas para evitar solapamientos o interferencias entre ellas. De este modo, cada canal ocupa una banda de frecuencia diferente y todas las señales pueden transmitirse simultáneamente por el mismo medio.

Multiplexación por División de Tiempo (TDM)

La multiplexación por división de tiempo consiste en subdividir el tiempo de transmisión en intervalos o ranuras (slots). La velocidad del medio de transmisión debe ser mayor que la de cada canal individual para poder intercalar las señales de forma secuencial. Cada canal transmite sus datos en su ranura asignada, que puede quedar vacía en caso de no tener datos para enviar, pero permanece reservada para ese canal.

Multiplexación por División de Longitud de Onda (WDM)

Esta técnica es conceptualmente similar a la FDM, pero aplicada en el ámbito óptico. Utiliza diferentes longitudes de onda de la luz, que pueden ser generadas por LEDs o láseres, para transmitir múltiples señales a través de la misma fibra óptica. La modulación consiste en emitir o no fotones en una longitud de onda determinada. Las longitudes de onda utilizadas en la luz visible están en el rango de nanómetros, mientras que las microondas tienen longitudes de onda del orden de centímetros.

5.4. Teoremas de Nyquist y Shannon

Teorema de Nyquist: Velocidad de Transmisión sin Ruido

El Teorema de Nyquist establece que la velocidad máxima de transmisión de información a través de un canal sin ruido está dada por la fórmula:

$$C = 2B \log_2(M) \quad (2)$$

donde:

- C es la velocidad de transmisión en bits por segundo (bps).
- B es el ancho de banda del canal, medido en Hertz (Hz).
- M es la cantidad de niveles o símbolos que se pueden transmitir.

Un caso particular es cuando se utilizan dos niveles, es decir, $M = 2$. En ese caso, la fórmula se simplifica a:

$$C = 2B \quad (3)$$

Esto significa que la velocidad máxima es dos veces el ancho de banda del canal.

Ejemplo: Consideremos un módem telefónico con un ancho de banda de 4 KHz. Aplicando el Teorema de Nyquist con $M = 2$, la velocidad máxima de transmisión será:

$$C = 2 \times 4000 = 8000 \text{ bps}$$

Por lo tanto, el módem puede transmitir hasta 8000 bits por segundo en condiciones ideales sin ruido.

Teorema de Shannon: capacidad del canal con ruido

El Teorema de Shannon establece que la capacidad máxima de un canal con ruido está dada por la fórmula:

$$C = B \log_2(1 + SNR) \quad (4)$$

donde:

- C es la capacidad del canal en bits por segundo (bps).
- B es el ancho de banda del canal, en Hertz (Hz).
- SNR es la relación señal a ruido, expresada en veces (no en decibelios).

Esta fórmula indica que la capacidad máxima del canal depende tanto del ancho de banda como del efecto del ruido en la señal. Es la capacidad máxima que puede alcanzar el receptor después de que la señal ha pasado por un canal con ruido.

Ejemplos:

1. Calcular la capacidad de un módem telefónico con $SNR = 20$ dB.

Recordemos que 20 dB equivale a una relación señal a ruido de 100 veces. Si el ancho de banda es $B = 4000$ Hz, entonces:

$$C = 4000 \times \log_2(1 + 100) = 4000 \times 6,65 = 26,632 \text{ bps} \approx 26 \text{ Kbps}$$

2. Si la empresa telefónica mejora la tecnología y la SNR aumenta en 10 dB, entonces $SNR = 30$ dB, que equivale a 1000 veces. Así:

$$C = 4000 \times \log_2(1 + 1000) = 4000 \times 9,96 = 39,868 \text{ bps} \approx 39 \text{ Kbps}$$

Ruido, Ruido Térmico y Densidad Espectral de Potencia de Ruido

En las comunicaciones a través de medios físicos, existen diversas fuentes de ruido que afectan la calidad de la señal transmitida. Entre ellas se encuentran el ruido térmico, el ruido impulsivo, interferencias electromagnéticas y de radiofrecuencia (RF), entre otros.

La *potencia de ruido térmico* está asociada a la agitación térmica de los electrones en los conductores y se calcula con la fórmula:

$$N = k \cdot T \cdot B \quad [W] \quad (5)$$

donde:

-
- N es la potencia de ruido térmico en vatios (W).
 - k es la constante de Boltzmann, cuyo valor es $1,38 \times 10^{-23}$ [J/K].
 - T es la temperatura absoluta en grados Kelvin (K).
 - B es el ancho de banda del canal, en Hertz (Hz).

Además, se define la *densidad espectral de potencia de ruido* como la potencia de ruido por unidad de ancho de banda, que se expresa como:

$$N_0 = \frac{N}{B} = k \cdot T \quad [W/Hz] \quad (6)$$

Por lo tanto, la potencia de ruido total se puede escribir también como:

$$N = N_0 \cdot B \quad (7)$$

Esto indica que la potencia de ruido térmico es directamente proporcional tanto a la temperatura del sistema como al ancho de banda del canal.

Referencias

- [1] Kurose, J. F., & Ross, K. W. (2016). *Computer Networking: A Top-Down Approach* (7th ed.). Pearson.