

# Módulo 01

## Introducción a las Redes de Computadoras (Pt. 2)



Redes de Computadoras  
Depto. de Cs. e Ing. de la Comp.  
Universidad Nacional del Sur



# Copyright

- Copyright © **2010-2022** A. G. Stankevicius
- Se asegura la libertad para copiar, distribuir y modificar este documento de acuerdo a los términos de la **GNU Free Documentation License**, versión 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera
- Una copia de esta licencia está siempre disponible en la página <http://www.gnu.org/copyleft/fdl.html>
- La versión transparente de este documento puede ser obtenida de la siguiente dirección:

<http://cs.uns.edu.ar/~ags/teaching>



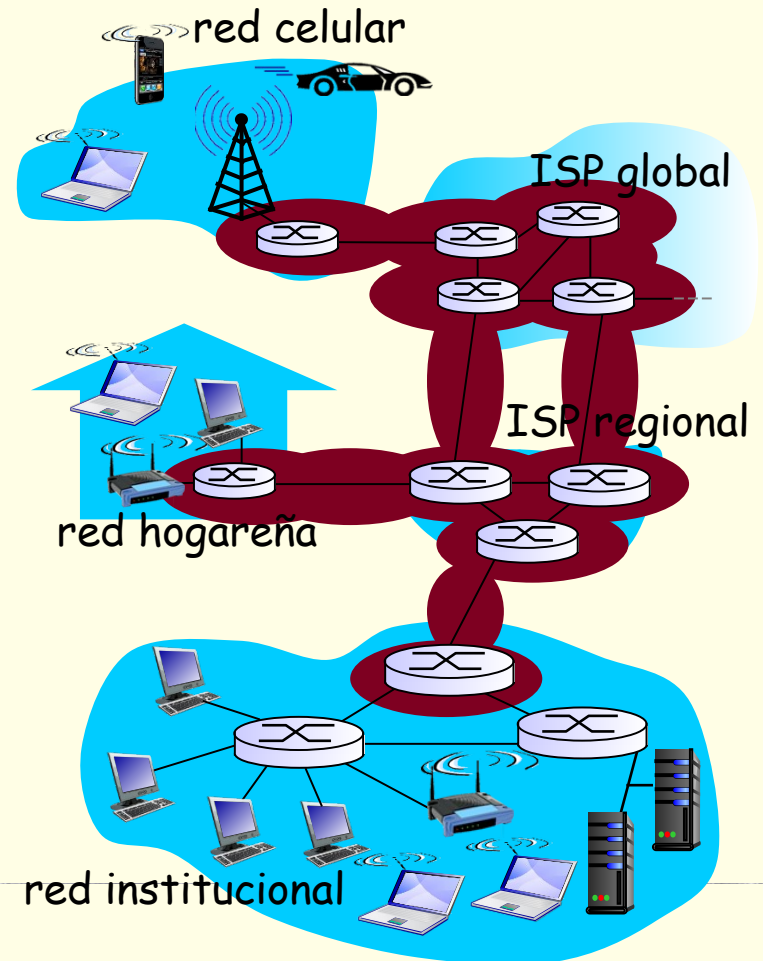
# Contenidos

- ¿Qué es internet?
- Concepto de protocolo
- Frontera y núcleo de una red
- Conmutación de circuitos y de paquetes
- Latencia y ancho de banda
- Organización de internet
- Modelos de referencia **ISO/OSI** y **TCP/IP**
- La red bajo constante amenaza



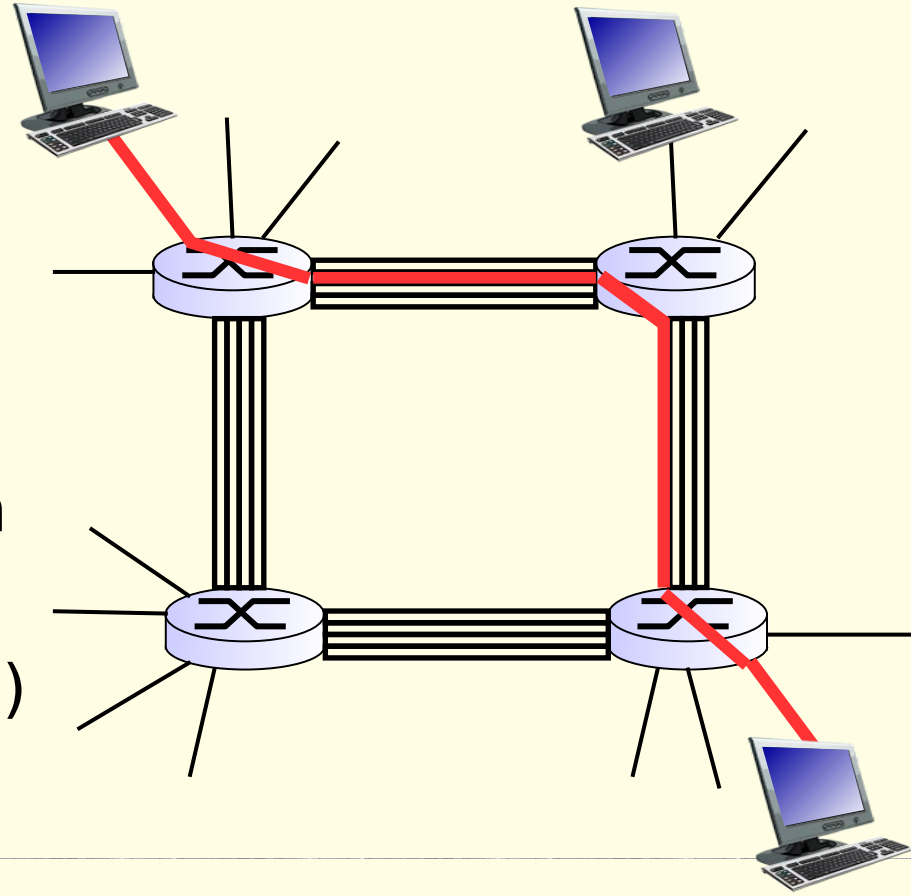
# Núcleo de red

- El núcleo de la red es un **reticulado de routers**
- La pregunta del millón:
  - ➔ ¿Cómo hacer para transferir datos a través de la red?
- A lo largo del tiempo se ensayaron dos soluciones:
  - ➔ La **conmutación de circuitos**
  - ➔ La **conmutación de paquetes**



# Conmutación de circuitos

- El ejemplo clásico es el viejo sistema telefónico
  - Crea un circuito físico entre los teléfonos
- Se reservan recursos de punta a punta:
  - **Uso exclusivo** de los recursos, no se comparten
  - **Desempeño garantizado** (similar a tener un circuito)
  - **Requiere inicialización**



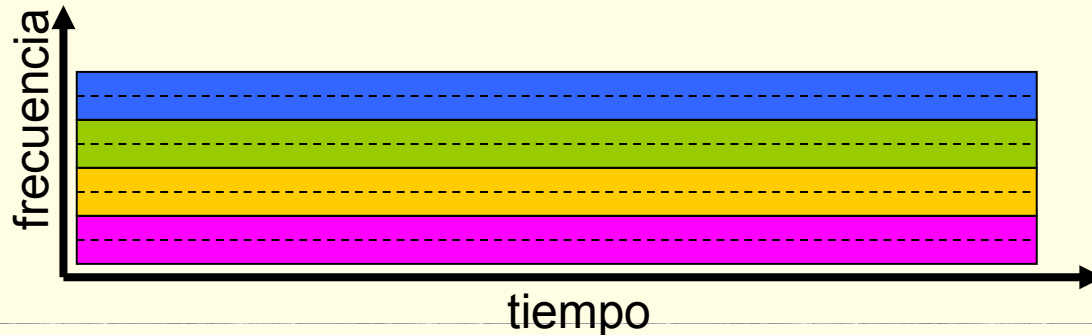
# Conmutación de circuitos

- Los recursos de la red deben dividirse en pequeñas porciones que luego serán asignadas a los distintos usuarios
  - Cada porción es asignada a una “llamada” diferente
- El **ancho de banda** de los enlaces (esto es, su capacidad para transportar información) debe poder cortarse en pequeñas porciones:
  - Multiplexado por **división de frecuencia**
  - Multiplexado por **división de tiempo**



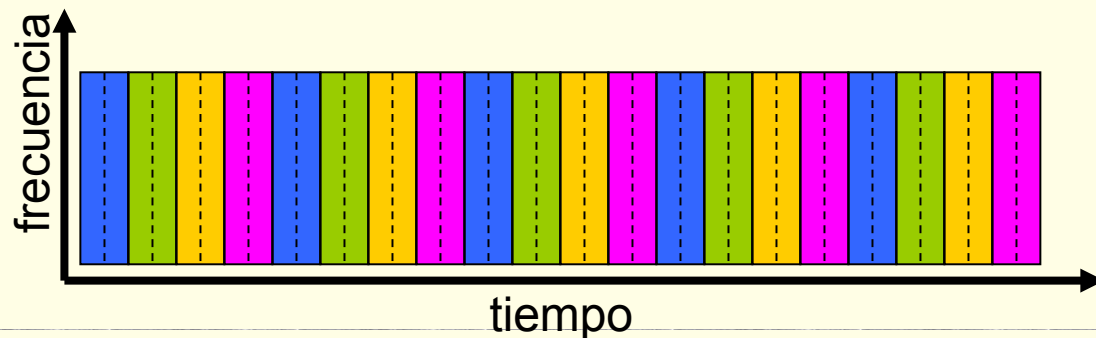
# División por frecuencia

- El multiplexado por división de frecuencia (**FDM**) consiste en asignar una frecuencia diferente a cada usuario
- Gráficamente, para cuatro usuarios:



# División por tiempo

- El multiplexado por división de tiempo (TDM) consiste en asignar una porción de tiempo diferente a cada usuario, de manera cíclica
- Gráficamente, para cuatro usuarios:





# Ejemplo

- Supongamos que se desea transmitir un documento de **640.000 bits** por una red de conmutación de circuitos
  - La velocidad de todos los enlaces es de **1,536 Mbps**
  - Los enlaces usan multiplexado **TDM** de **24 slots**
  - Asumir que el establecimiento de la conexión insume **500ms**
- En este contexto, ¿cuánto tiempo tarda la transmisión del documento?
  - A sacar la calculadora, el celu o si no el ábaco...



# Conmutación de paquetes

- Se observó que los usuarios hacen un uso muy particular de la red:
  - ➔ Usualmente la red está **totalmente ociosa**, si bien esporádicamente aparece una **ráfaga de uso intenso**
- En este escenario, la conmutación de circuitos **desaprovecha ancho de banda**
- La solución obvia es compartir los enlaces de una manera más eficiente:
  - ➔ Cortar los datos en pequeños pedazos (llamados **paquetes**) y conmutar en cambio esos paquetes



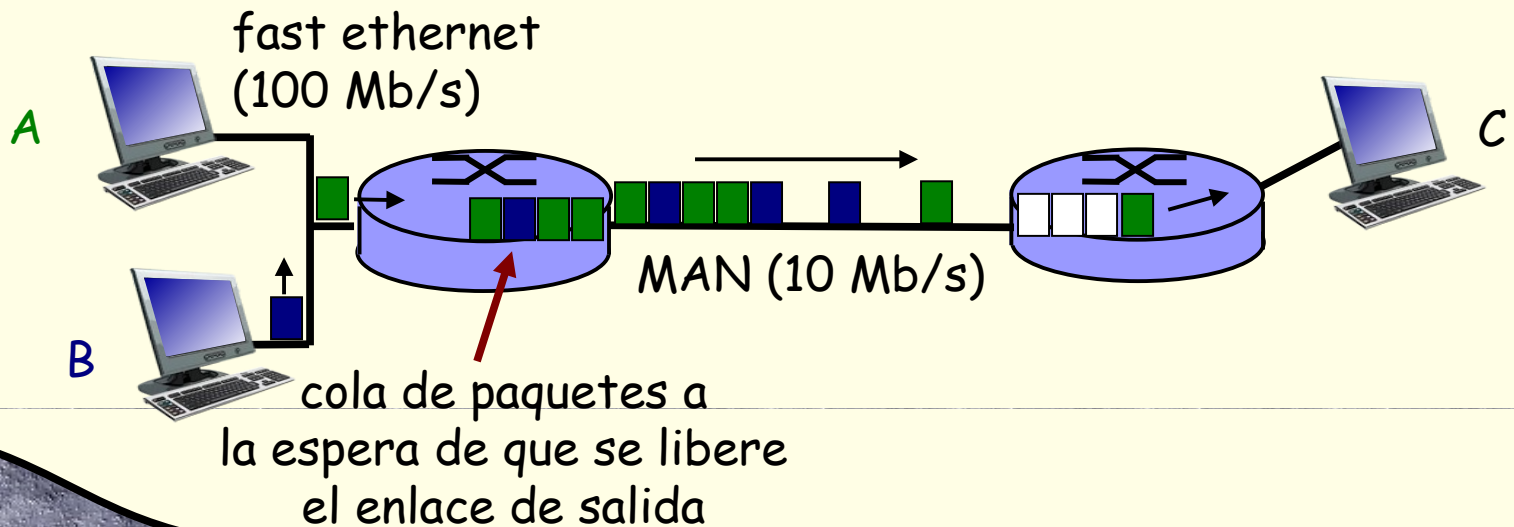
# Conmutación de paquetes

- La clave de la conmutación de paquetes es que los enlaces no son más exclusivos, los usuarios ahora pueden compartirlos
  - Cuando un usuario tiene un paquete para enviar, **hace uso de la totalidad del ancho de banda**
  - Los recursos son asignados según hagan falta
  - No es necesario reservar previamente recursos ni acordar cómo dividir el ancho de banda
  - Puede haber contención en el uso de los recursos, lo que resulta en una **congestión de tráfico**



# Paquetes vs. TDM

- La conmutación de paquetes parece adoptar una estructura análoga al multiplexado por división de tiempo
- No obstante, existe una diferencia fundamental en este caso, pues la conmutación **no configura un patrón cíclico preestablecido**:



# Circuitos vs. Paquetes

- Supongamos un conjunto de usuarios con las siguientes características:
  - Cada uno requiere **1Mb/s** cuando está activo
  - Están activos usando la red el **10%** del tiempo
- ¿Qué es más conveniente conmutar si cuento con un enlace de **5Mb/s**?
  - Si conmutamos circuitos, con 5 usuarios **se agota el ancho de banda**
  - Si conmutamos paquetes, con 20 usuarios el **99.12%** del tiempo hay 6 o menos usuarios activos



# Circuitos vs. Paquetes

- A la luz de la evidencia, la conmutación de paquetes surge como claro ganador... ¿o no?
  - ➔ La conmutación de paquetes es ideal para tráfico de tipo ráfaga
  - ➔ Se comparten los recursos de red y no requiere inicialización de antemano
- No obstante, ino todo lo que brilla es oro!
  - ➔ Se puede dar una congestión y los protocolos también deben contemplar la eventual pérdida de paquetes
  - ➔ No permite asegurar la calidad del servicio



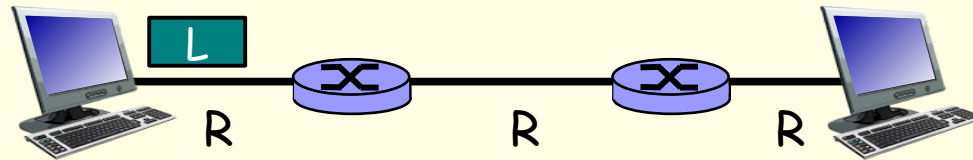
# Ancho de banda y latencia

- Al considerar la calidad de un enlace el primer parámetro que notamos es su **ancho de banda**
  - ➔ En computación, denominaremos ancho de banda a la capacidad ideal de transferencia de información por unidad de tiempo de un determinado enlace o dispositivo
- Sin embargo, la conmutación paquetes puede causar congestiones por lo que también debemos tener en cuenta la **latencia**
  - ➔ Denominaremos latencia al tiempo que le toma a un cierto dato en atravesar la red



# Store and forward

- Nótese que hacen falta  $L/R$  segundos para empujar los  $L$  bits de un paquete dentro de un enlace con un ancho de banda  $R$
- A su vez, el paquete tiene que ser recibido completamente antes de comenzar a ser enviado a través del próximo enlace
- Esta operatoria se denomina “store and forward”



retardo total:  $3L/R$

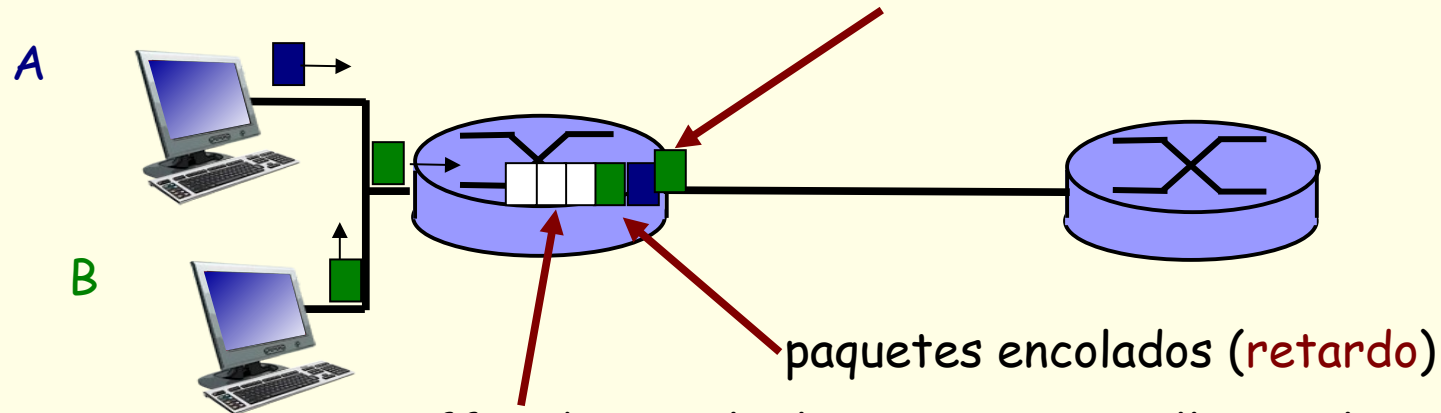
(ignorando otras fuente de retardo)





# Pérdidas y retardos

- Los paquetes en tránsito a veces terminan encolados en los buffers de los routers
  - Llegan más paquetes de los que podemos enviar por el enlace de salida
  - Los paquetes deben esperar que les toque su turno
    - paquete que está siendo transmitido (**retardo**)

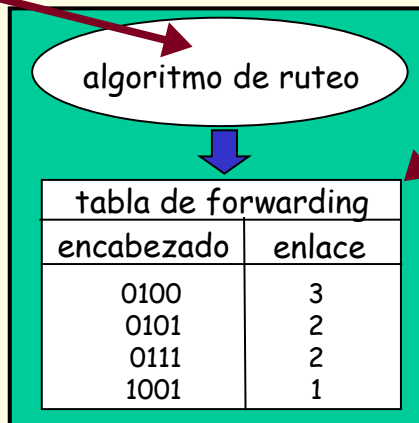


buffer disponible: los paquetes que llegan al router son descartados al agotarse el buffer disponible (**pérdida**)

# Dos funciones esenciales

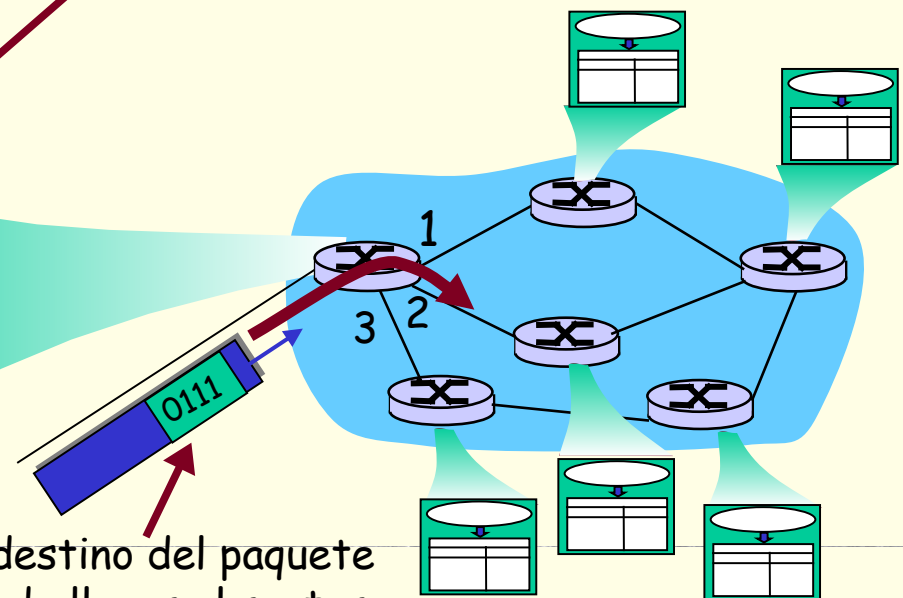
## ● Ruteo:

- Consiste en encontrar el camino del origen al destino
- Para esto se usan los algoritmos de ruteo



## ● Forwarding:

- Se trata de asegurarse de que el paquete salga por el enlace que debe



dirección destino del paquete que acaba de llegar al router



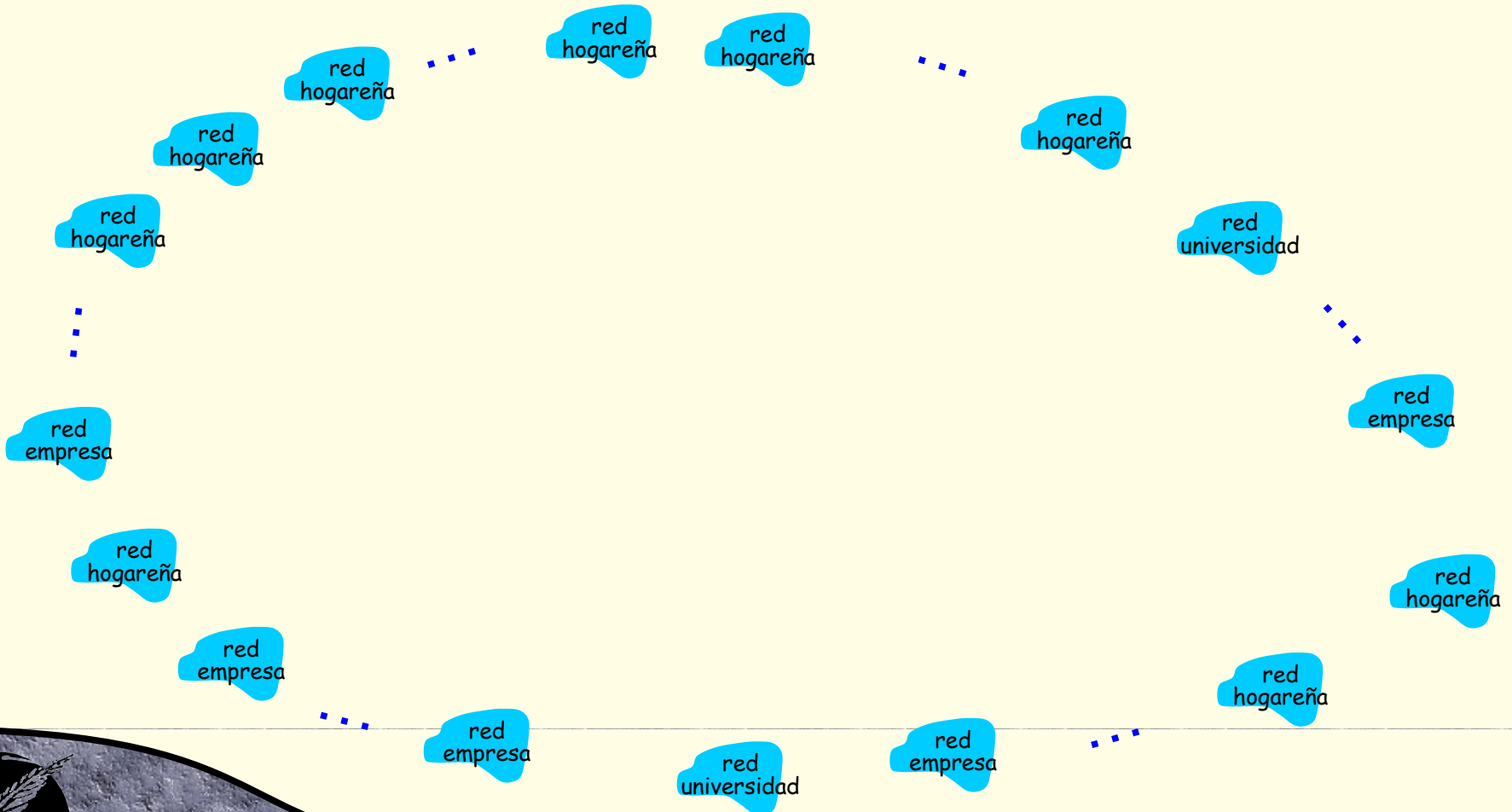
# Organización de internet

- En internet la frontera se conecta al núcleo a través de los **ISPs** (Internet Service Provider)
- Naturalmente los **ISPs** necesitan estar interconectados entre sí
  - En los albores... ino era así!
- La topología resultante terminó siendo muy compleja
  - La evolución no fue del todo planeada, se vio afectada por cuestiones económicas y también de soberanía



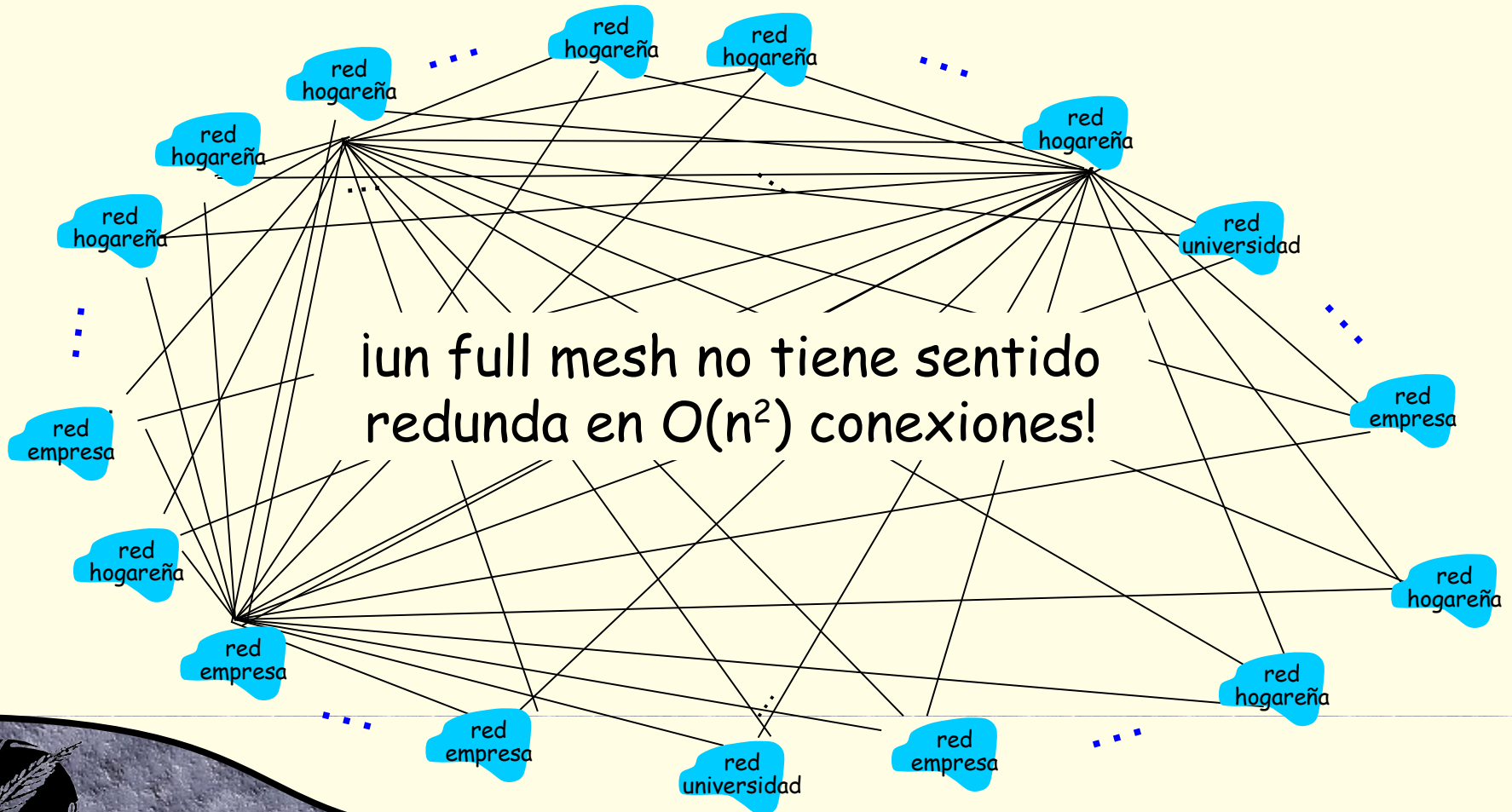
# Organización de internet

• ¿Cómo conectar millones de redes entre sí?



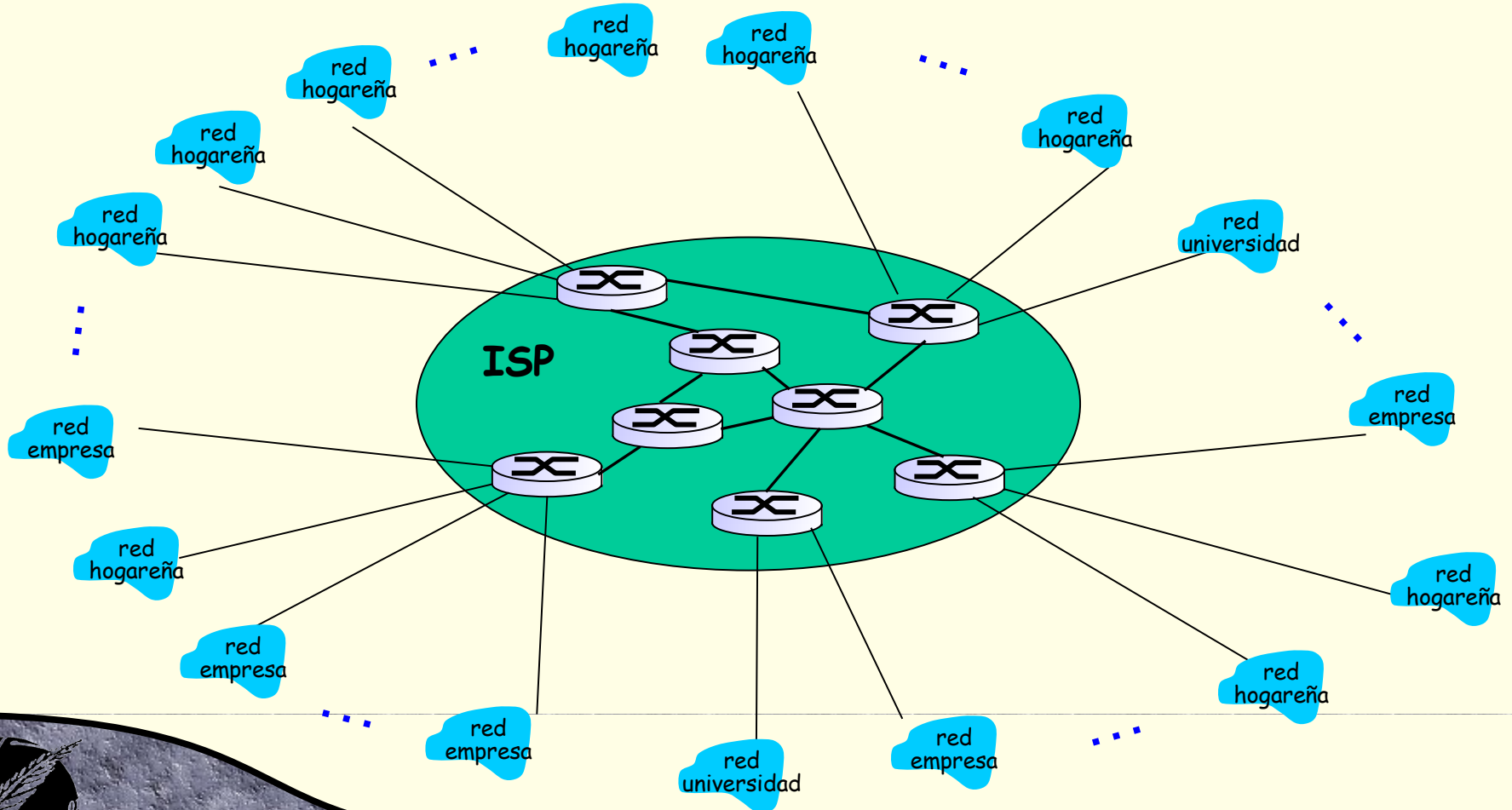
# Organización de internet

• ¿Todos con todos? (un full mesh)



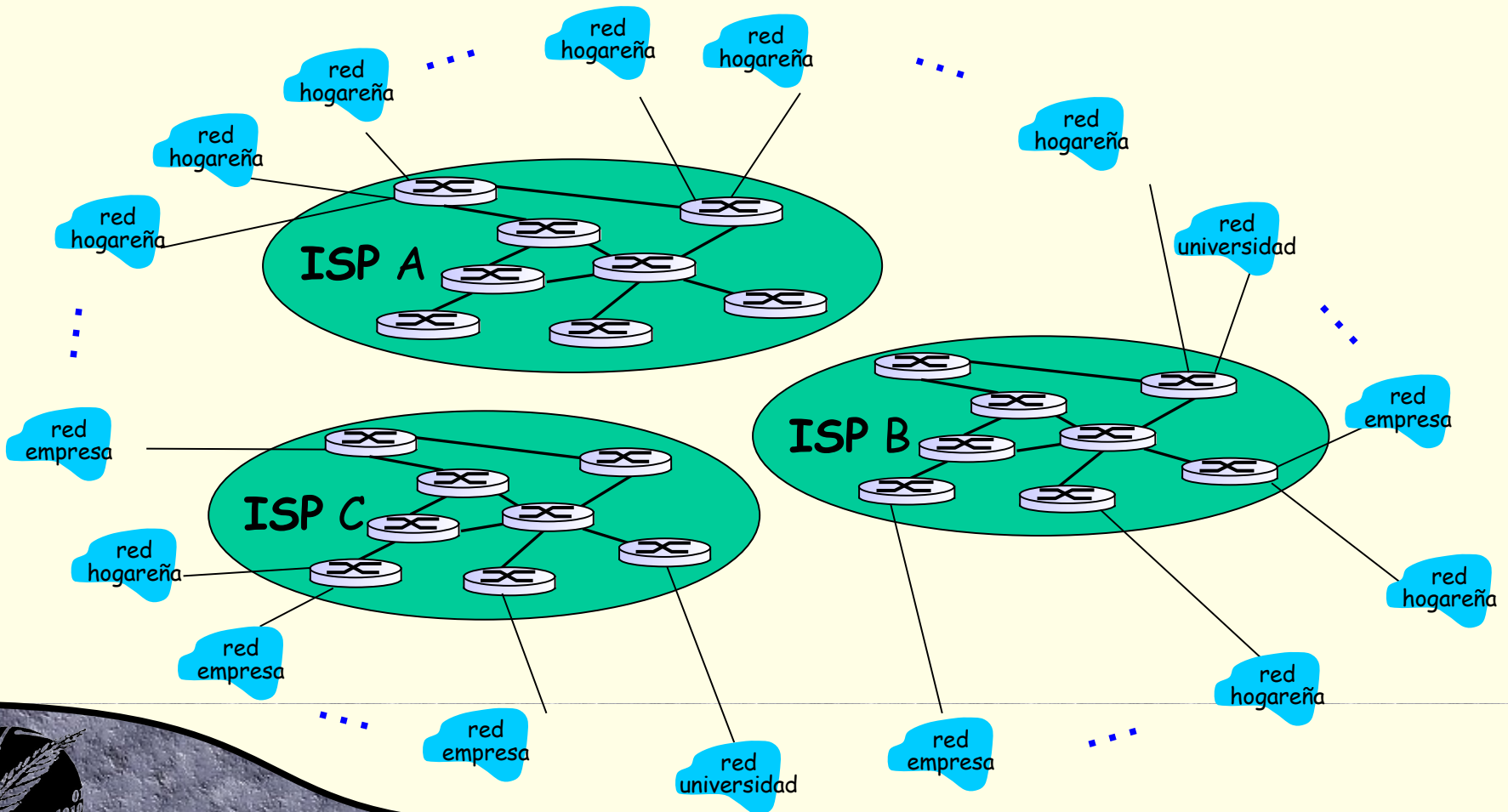
# Organización de internet

• ¿Qué tal conectar las redes a un proveedor?



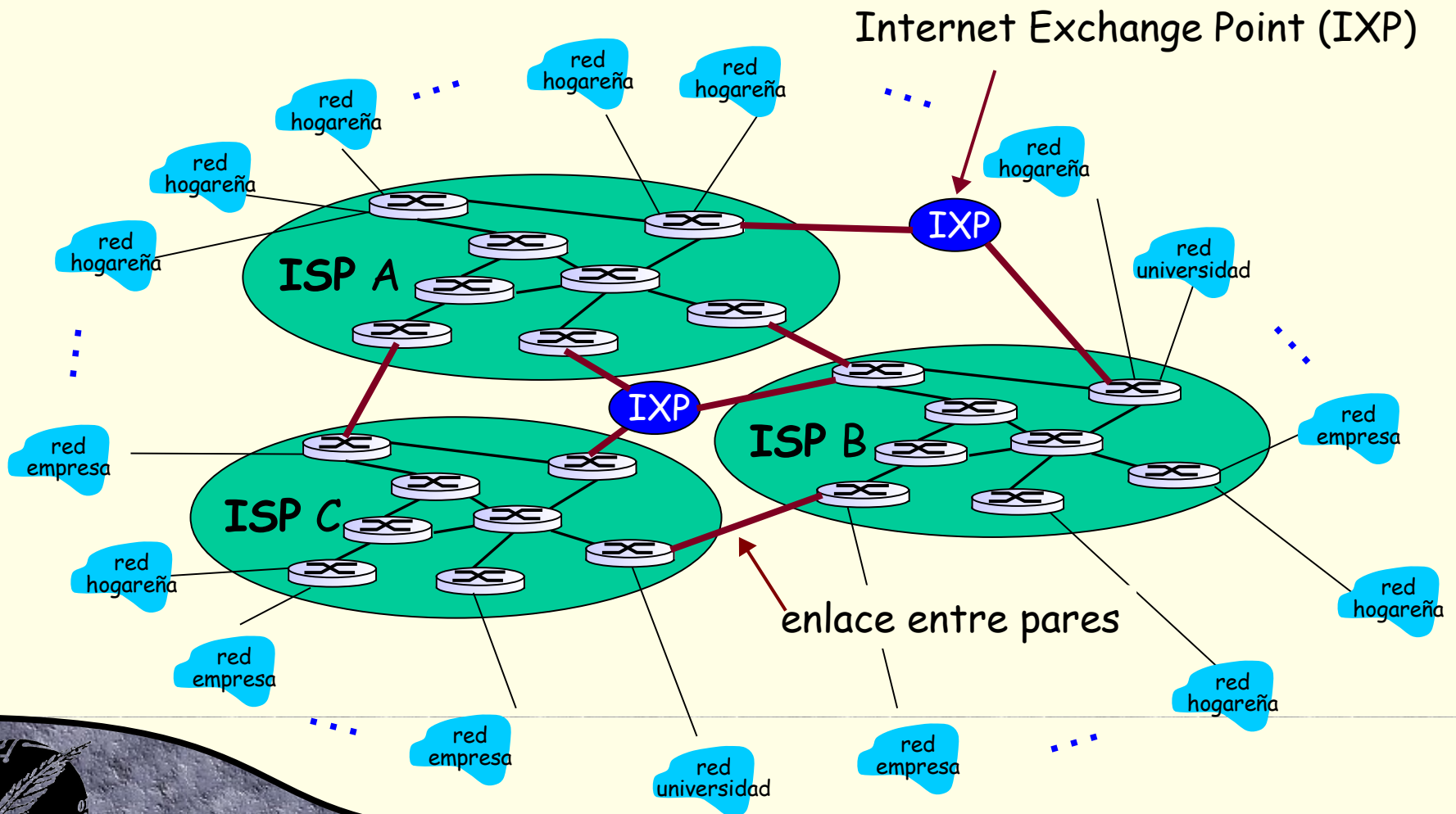
# Organización de internet

● Si la idea es buena... isurge la competencia!



# Organización de internet

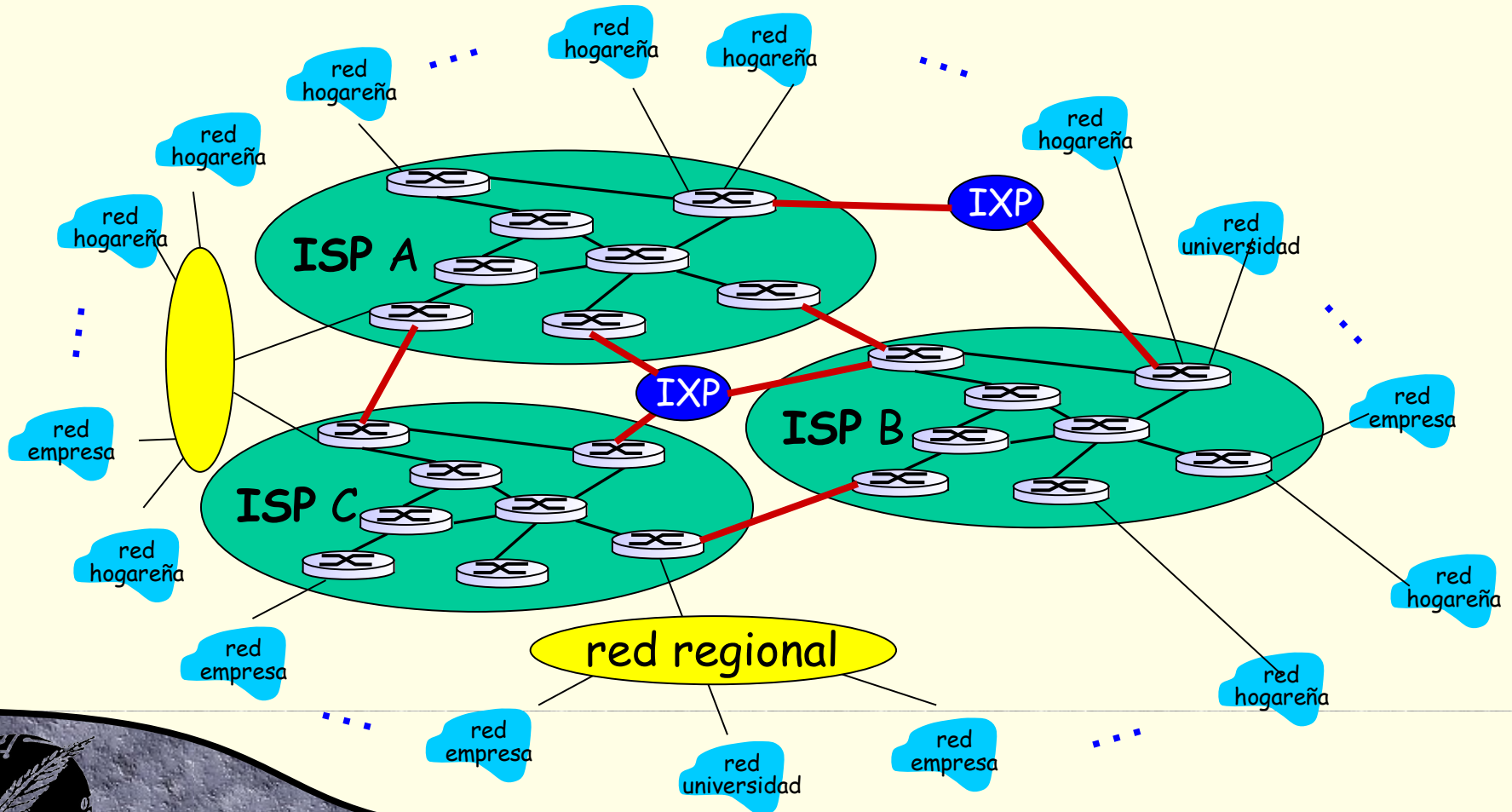
- Los competidores necesitan estar conectados





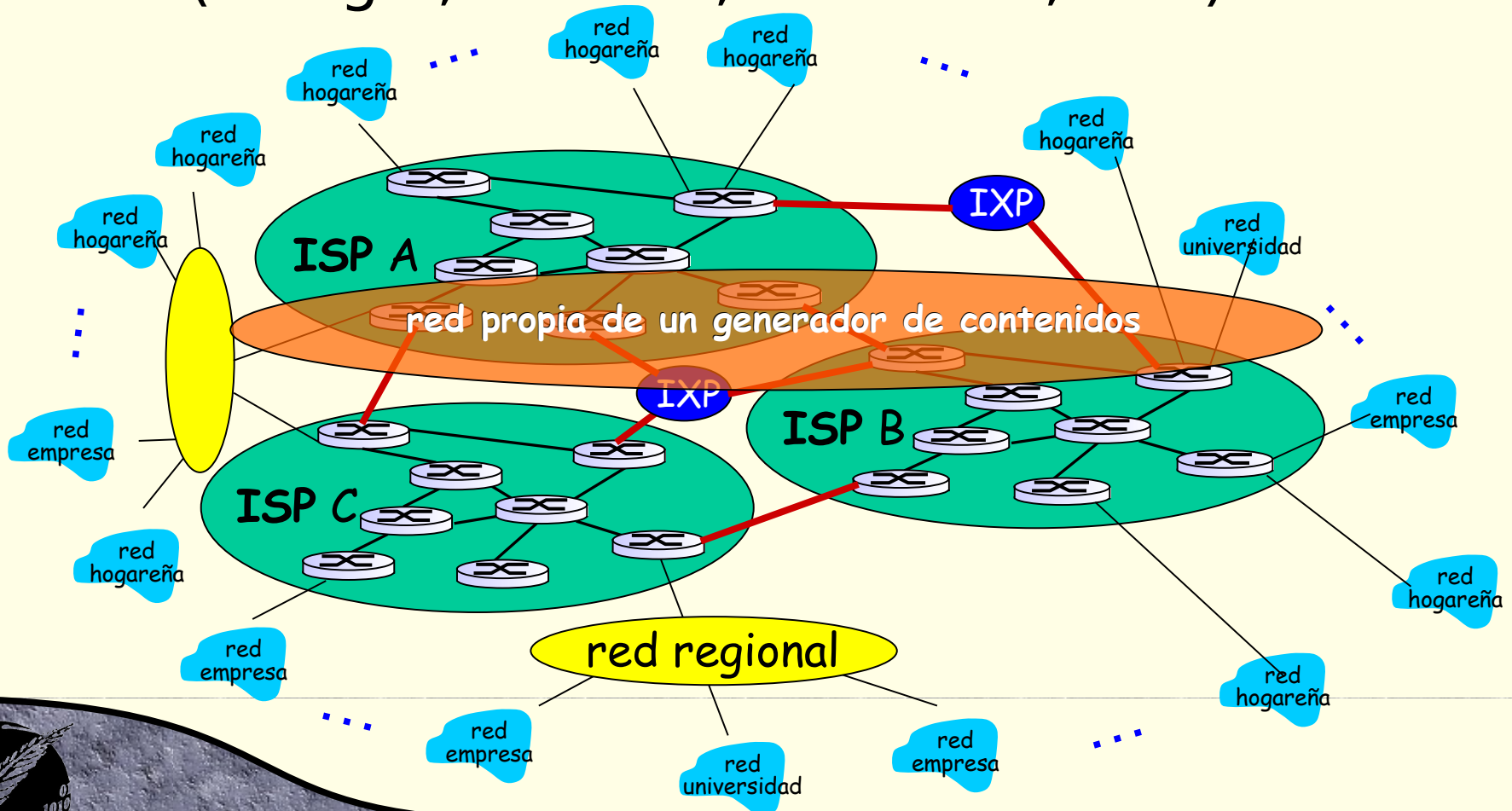
# Organización de internet

• A esta altura pueden aparecer redes regionales



# Organización de internet

- Los generadores de contenido tiene sus propias redes (Google, Akamai, Microsoft, etc.)



# Organización de internet

- ¿Qué motiva a compañías como Google o Facebook a mantener semejantes redes?
  - ➔ En general, el camino entre cualesquiera dos nodos atravesará múltiples nodos
  - ➔ La operatoria store and forward hace que los retardo de cada nodo se vayan acumulando
  - ➔ Las compañías mantienen redes privadas para brindar un mejor servicios a sus clientes
- ¿Cuánto tarda en resolver una consulta Google? ¿dónde está el servidor que respondió?



# ¿Preguntas?

