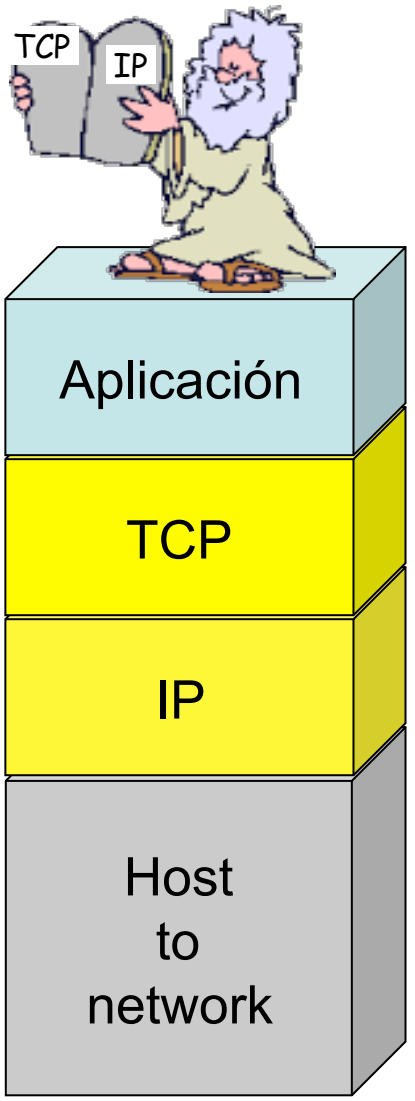
upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa
Redes de Nueva Generación
Área de Ingeniería Telemática

TCP/IP y Ethernet

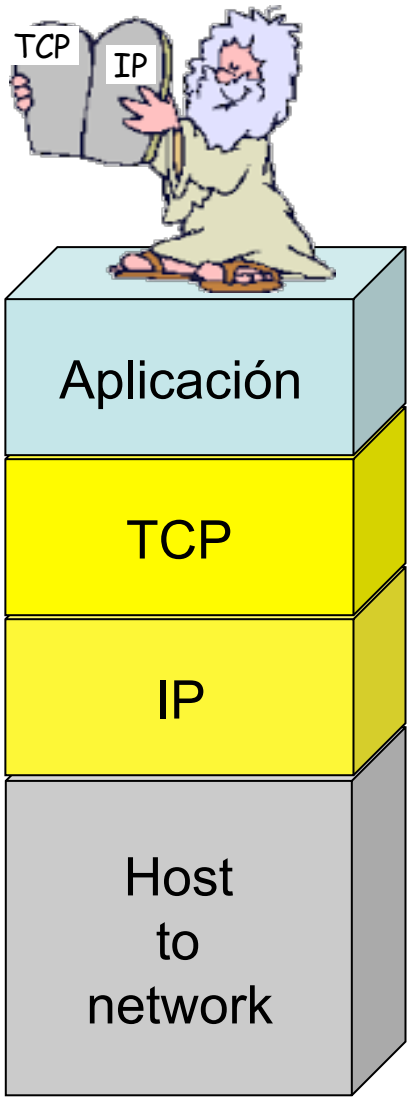
TCP/IP

- Vamos a hablar de Internet
- Eso quiere decir hablar de TCP/IP
- ¿TCP/IP cayó del cielo?

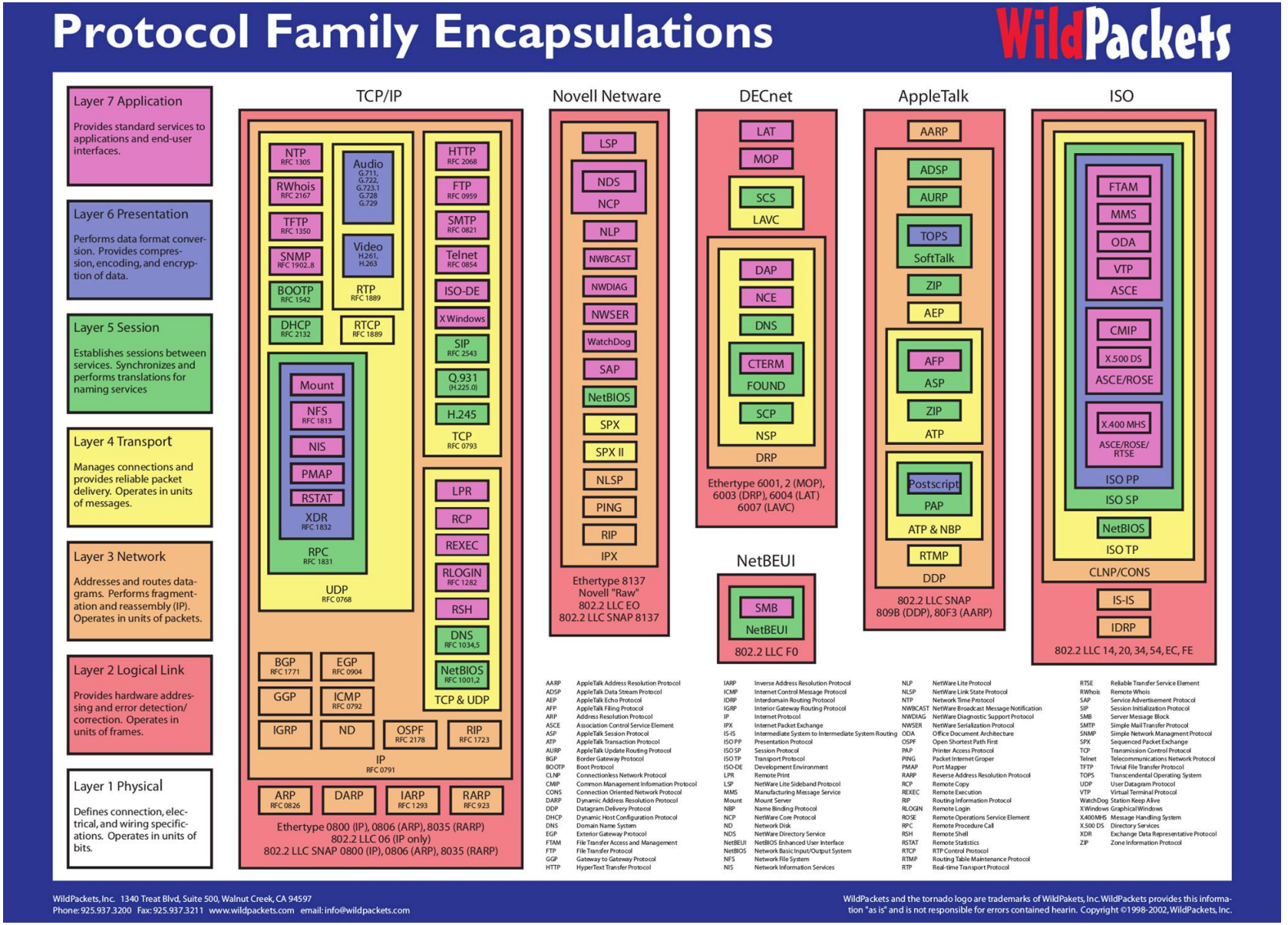


TCP/IP

- Hay otras arquitecturas de protocolos
- La arquitectura OSI
- IBM con SNA tenía la suya (que inspiró OSI)
- También Apple
- Novell
- Digital
- (…)

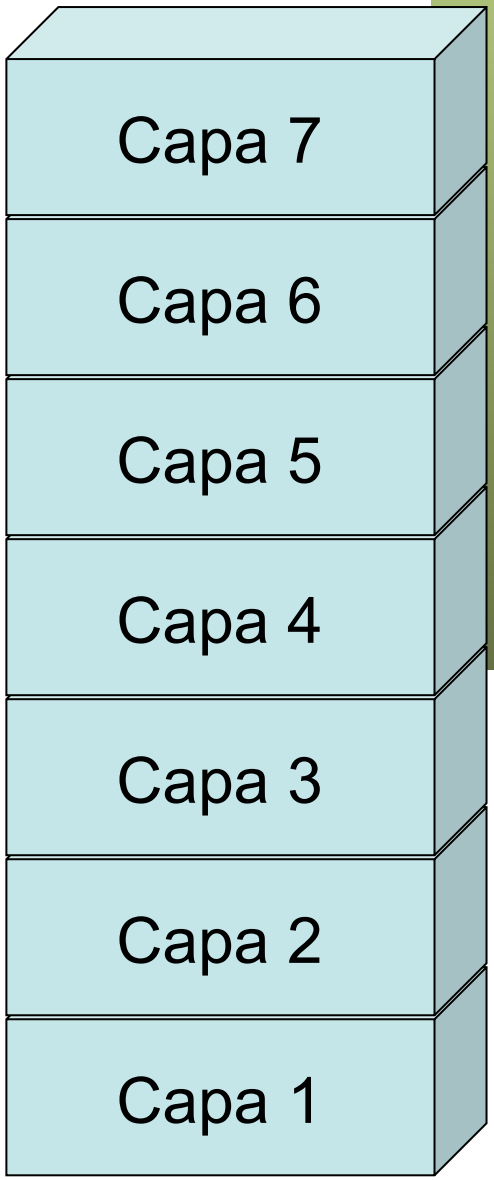


Familias de protocolos



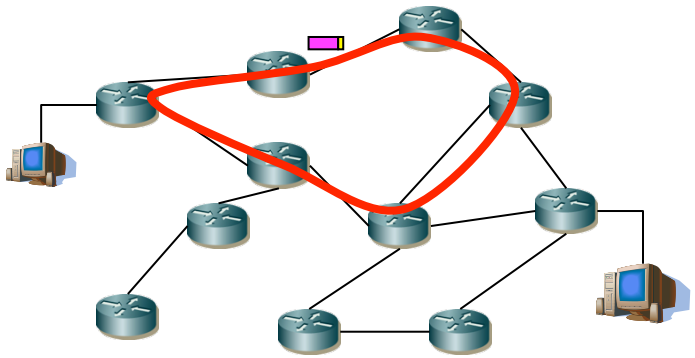
OSI

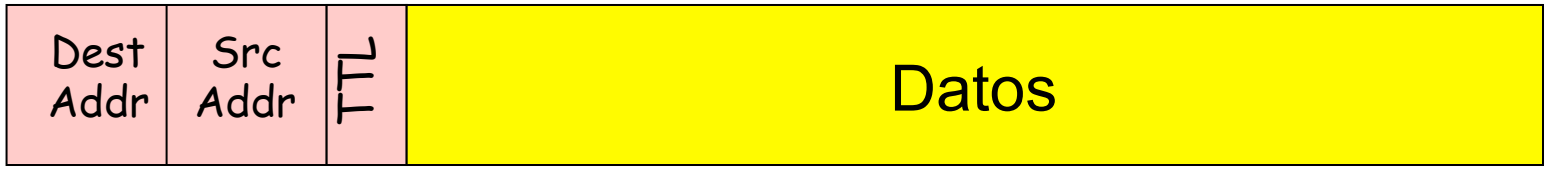
- Layer 1: envío de bits, conectores, medios físicos
- Layer 2: enviar un mensaje a un vecino, alguien en el mismo “enlace” (link)
- Layer 3: hay un “camino” a través de una “red” de origen a destino
- Layer 4: extremo a extremo, retransmisiones, reorden, etc



Layer 3: PDU básica

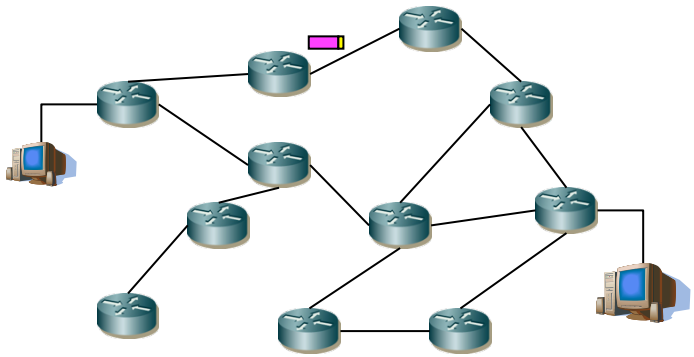
- Una dirección origen y destino
- Los datos
- Y también una cuenta de saltos
  - Algo que limite el número de saltos que dé el paquete, ¿por qué?
  - Con un algoritmo distribuido que calcula las rutas hay transitorios
  - Durante los cuales hay bucles
  - Y quieres que los paquetes atrapados en uno se descarten pronto

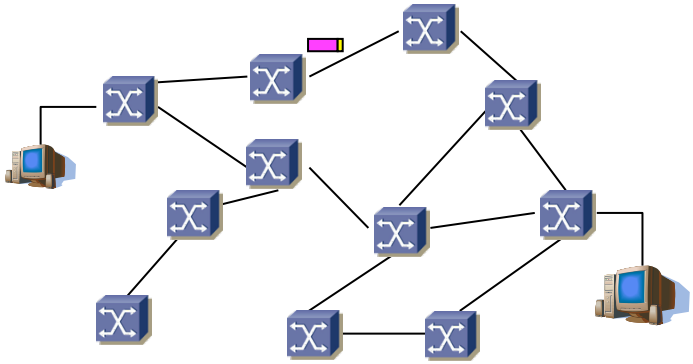




Ethernet

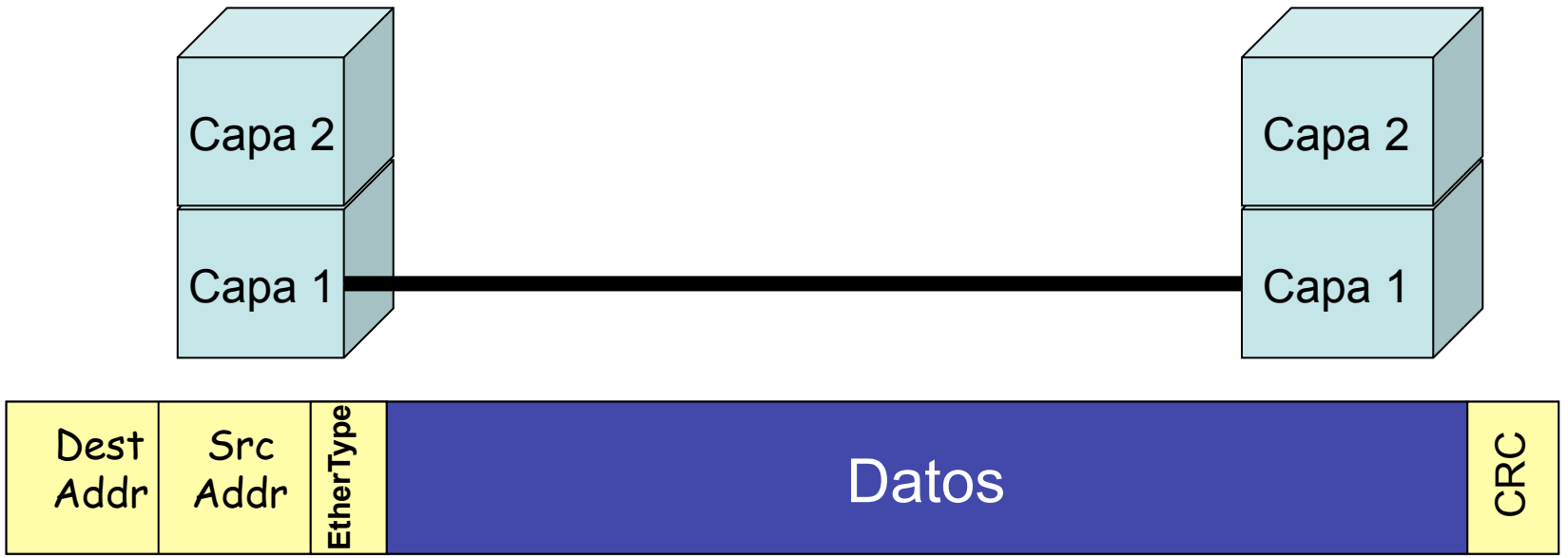
- Hoy en día en una LAN Ethernet tenemos conmutadores de paquetes
- Y la trama Ethernet se parece mucho a esto que hemos comentado
- Aunque no hay un TTL
- Y decimos que es una solución de capa 2
- ¿Por qué? ¿Cuál es exactamente la diferencia entre la capa 2 y la capa 3?





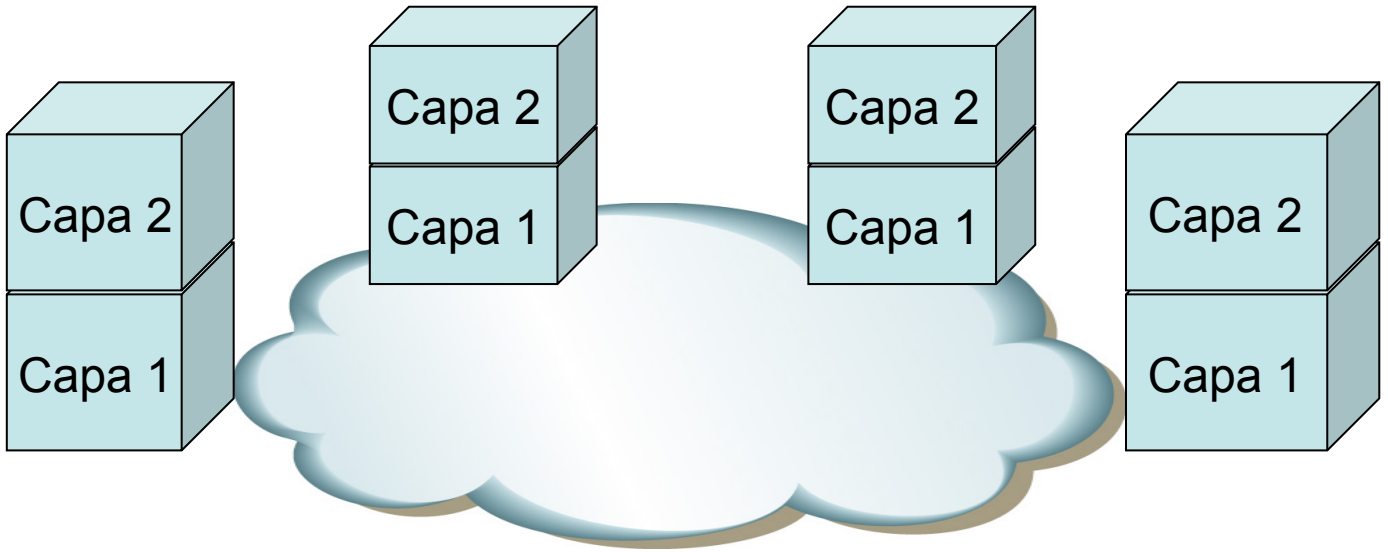
Ethernet

- Ethernet es para comunicar estaciones en el mismo enlace
- No se les ocurrió que alguien pudiera querer reenviar estos paquetes
- Por eso es de nivel de enlace



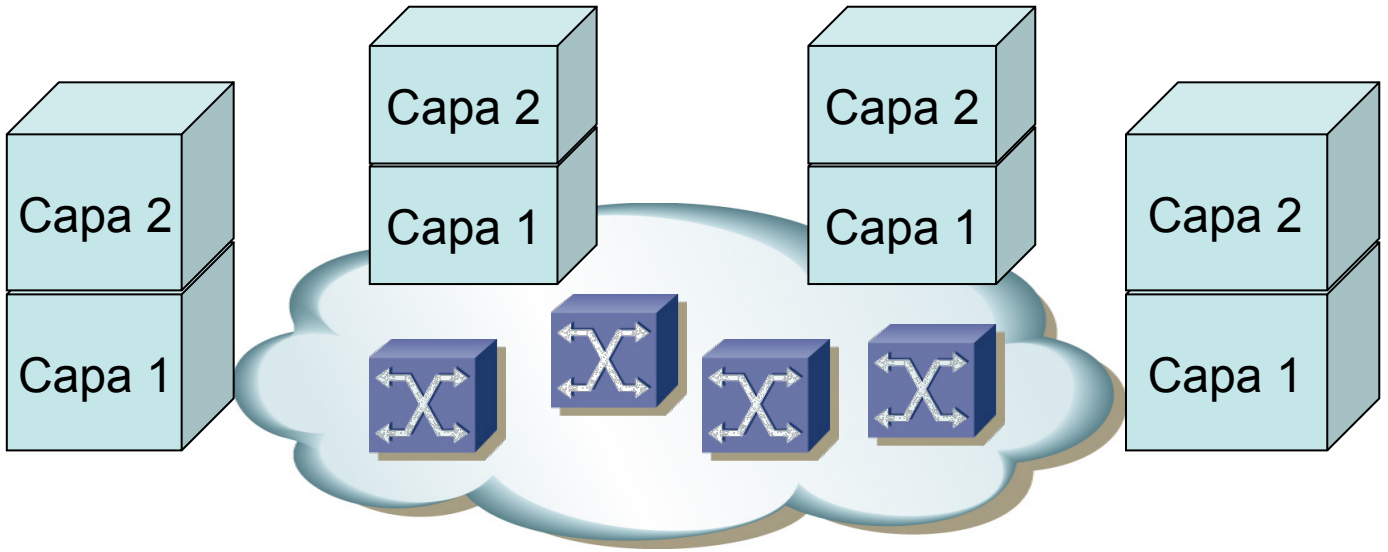
Ethernet es capa 2

- Pero la Ethernet original no era para un enlace, había una “Red de Área Local”, ¿no?
- Claro, el nivel de enlace resuelve el problema de estaciones que tienen un medio físico entre ellas (recordad el bus coaxial)
- Da por ejemplo el formato del mensaje
- O el protocolo de control de acceso al medio (CSMA/CD)
- La red de área local es ese segmento de coaxial o esos hubs



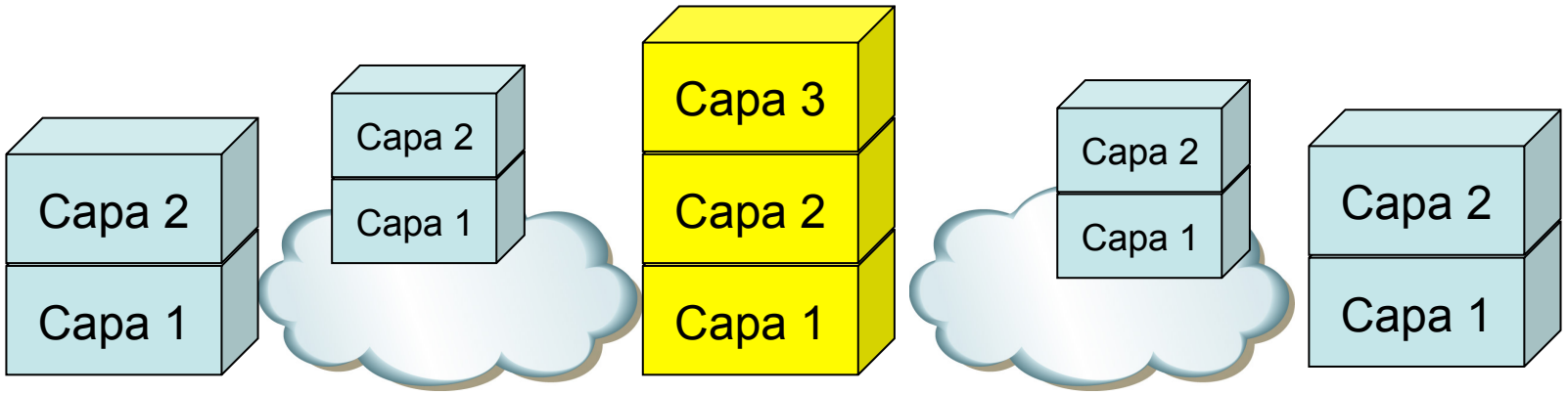
Ethernet es capa 2

- ¡ Pero esa red tiene conmutadores ! ¡ Haciendo almacenamiento y reenvío ! ¡ No me líes !
- ¿LAN Ethernet con conmutadores?
- Eso tiene de Ethernet 2 cosas:
  - El formato de la trama
  - El nombre
- La Ethernet como tal “murió” en los 90s con la introducción de los conmutadores



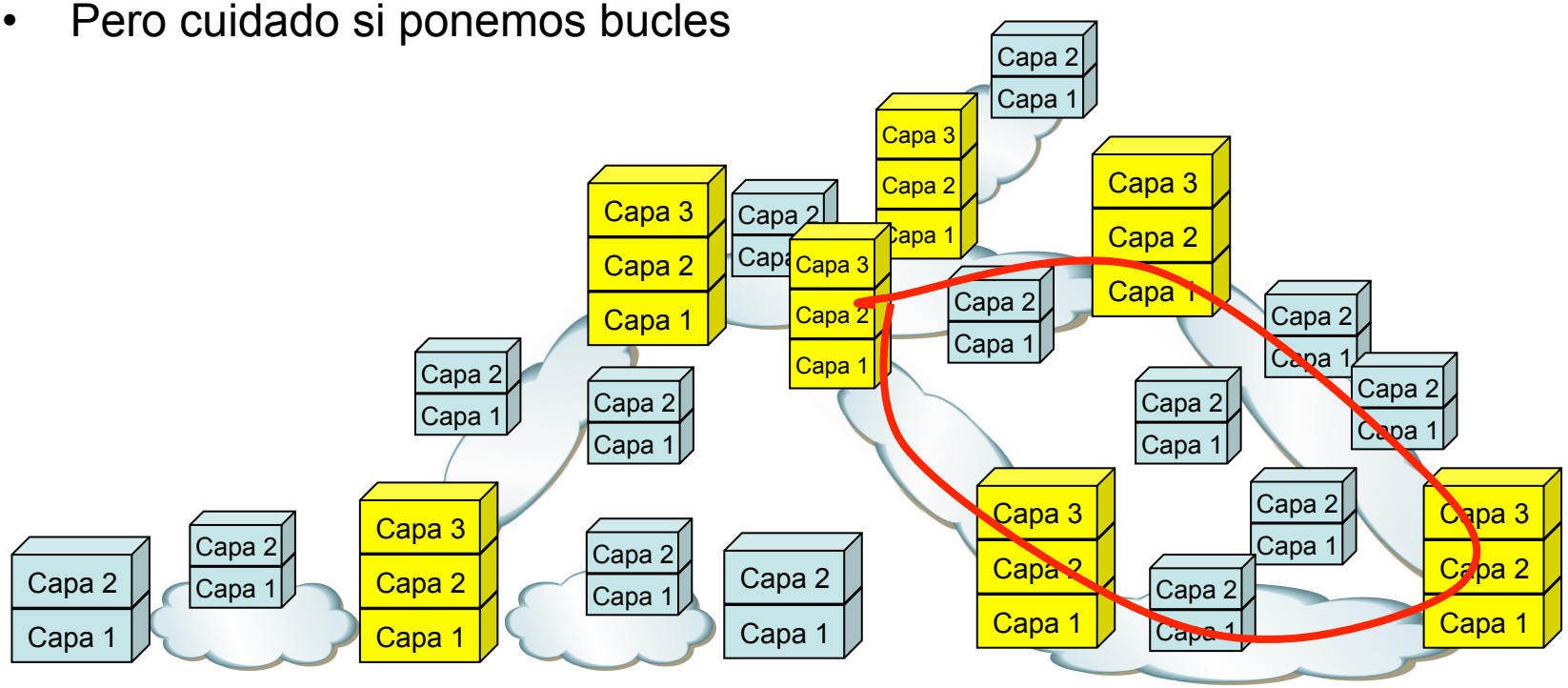
Ethernet vs Layer 3

- ¿Cómo "podría" ser la interconexión de LANs?
- En cada una de ellas tenemos el dominio de colisión Ethernet
- Interconectados por conmutadores capa 3
- Eso permitiría interconectar LANs Ethernet o con otras tecnologías
- Necesitaríamos el formato para el protocolo de capa 3



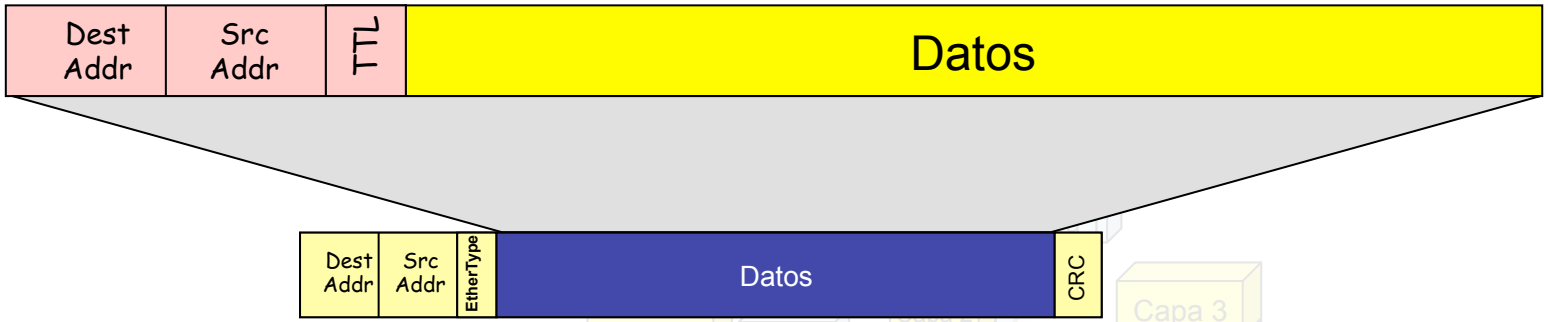
Ethernet vs Layer 3

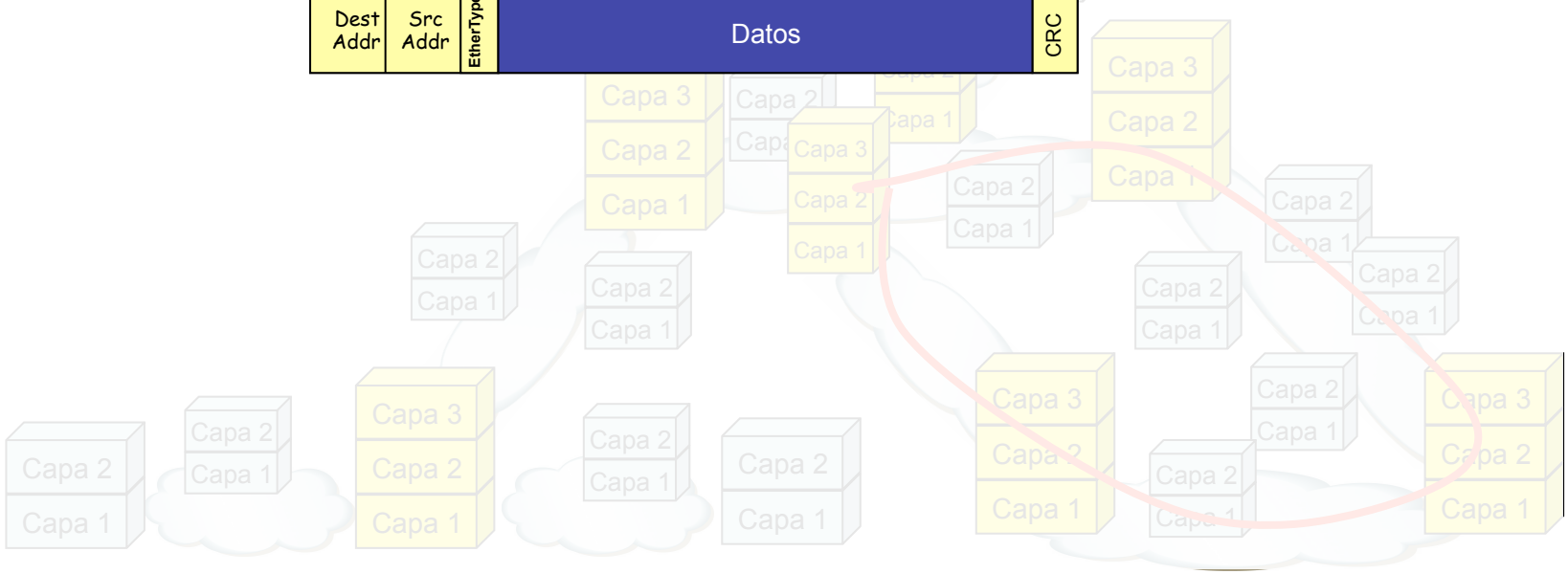
- ¿Cómo "podría" ser la interconexión de LANs?
- En cada una de ellas tenemos el dominio de colisión Ethernet
- Interconectados por conmutadores capa 3
- Eso permitiría interconectar LANs Ethernet o con otras tecnologías
- Necesitaríamos el formato para el protocolo de capa 3
- Y podríamos hacer más grande la topología pues no tenemos los límites de Ethernet
- Pero cuidado si ponemos bucles



Ethernet vs Layer 3

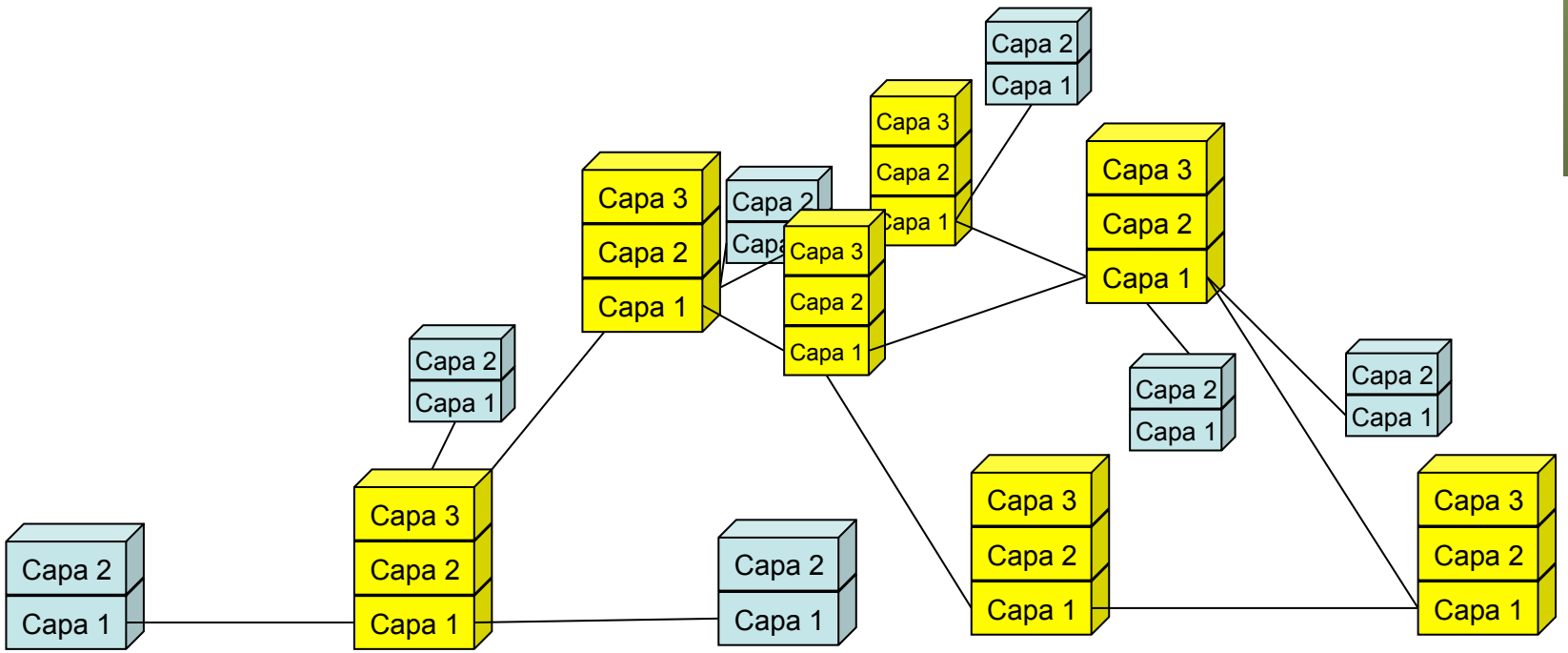
- Ok, necesitamos protocolos de encaminamiento dinámico
- Y mejor que le pongamos un TTL a esa PDU de capa 3
- Por supuesto esta PDU de capa 3 va dentro de la PDU de capa 2 en cada enlace
- O sea, en el caso Ethernet dentro de la trama Ethernet





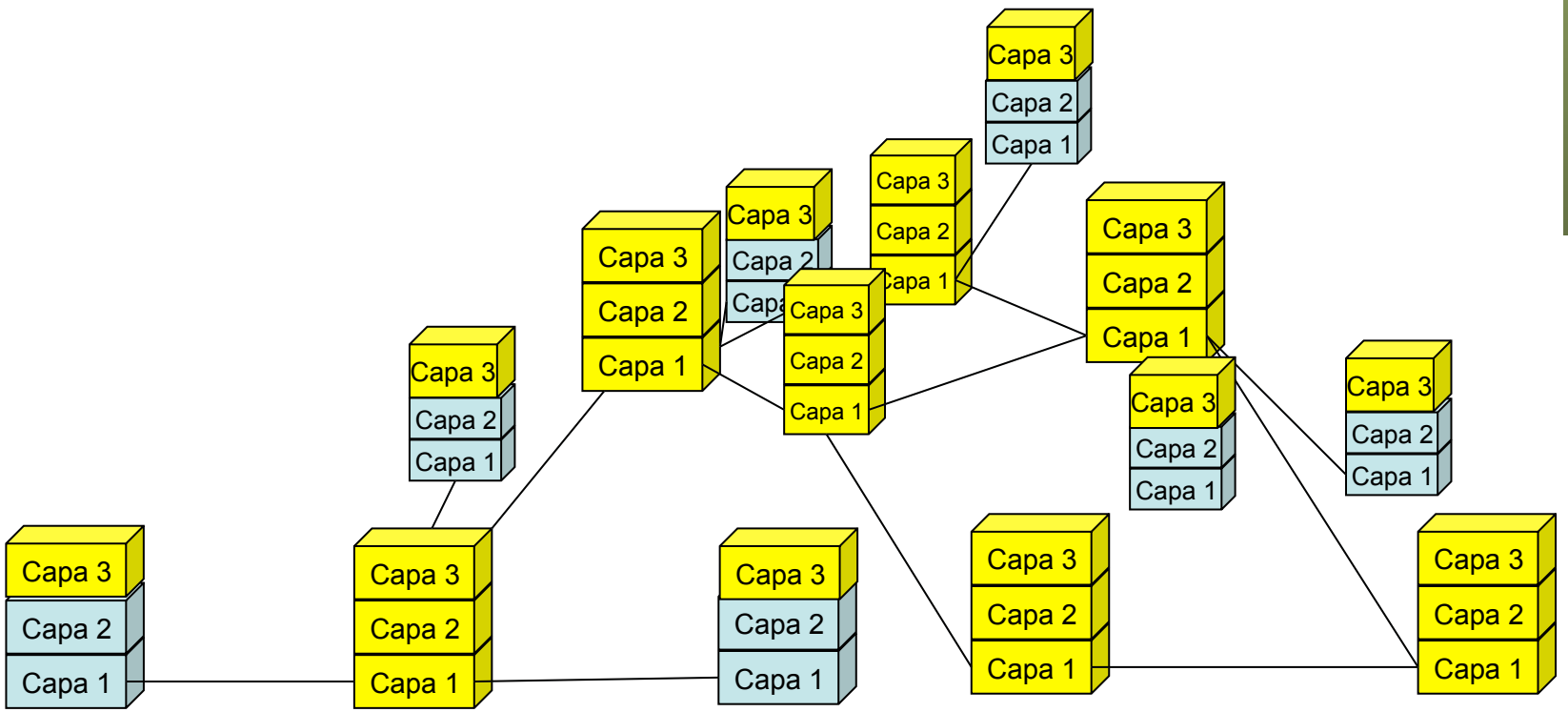
Ethernet vs Layer 3

- ¿Y es esto lo que tenemos hoy en día en las LANs Ethernet?
- Hemos reducido los dominios compartidos a un enlace con solo una estación
- Además el enlace es full-duplex con lo que no hace falta CSMA/CD
- ¿Los equipos son conmutadores capa 3?



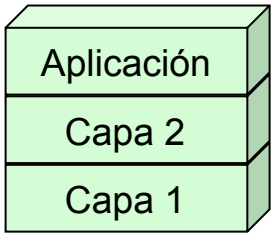
Ethernet vs Layer 3

- Mayormente los equipos son conmutadores capa 2
- ¿Por qué?
- En primer lugar porque si fueran capa 3 necesitaríamos el protocolo de capa 3 en los hosts



Ethernet vs Layer 3

- En los 80s-90s Ethernet tenía éxito
- Se quería mejorar su rendimiento
- Aislar los dominios de colisión lo permitiría
- ¿Con un conmutador de capa 3?
- El problema de un protocolo de capa 3 es que lo tienen que implementar los hosts
- No había un protocolo dominante en capa 3 (bueno, tampoco lo había todavía en capa 2 para LAN, pero eso es otro tema)



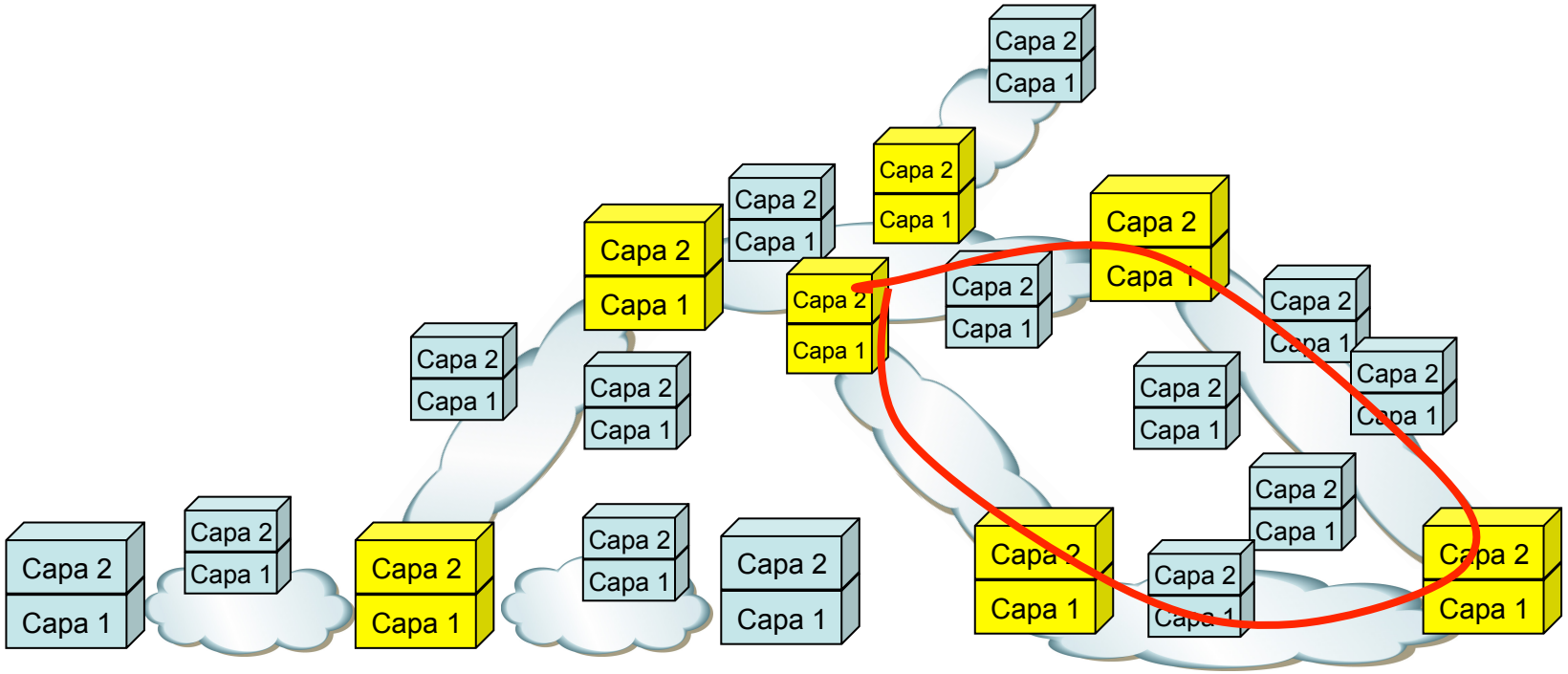
Ethernet vs Layer 3

- Podéis oír la historia por Radia Perlman:
  - https://www.youtube.com/watch?v=L_zacX9DcZA
- Trabajaba en la capa 3 (en los 80s en Digital)
- Pero entonces parecía que valía con la capa 2 de Ethernet
- Muchas aplicaciones se construían sobre la capa 2
- *"…and I said: but you may wanna talk from one Ethernet to another! And they said: our customers will never wanna do that"*
- *"…my manager says to me: Radia we need to design a magic box that will sit between two Ethernets and let somebody on one talk to somebody on another. Which is of course a router, but a router only works if the end-node is doing the same layer 3 protocol as the router"*
- *"We had to invent a box that was not allowed to modify the Ethernet packet in any way"*

- Y así nació el puente Ethernet (y con él más tarde el conmutador)

Ethernet vs Layer 3

- ¿Y los bucles?
- Tendríamos problemas con un protocolo de encaminamiento de capa 3 pues la trama Ethernet no tiene un TTL
- Así que se diseñó (Radia, en un par de días) un protocolo para eliminar esos ciclos
- (…)

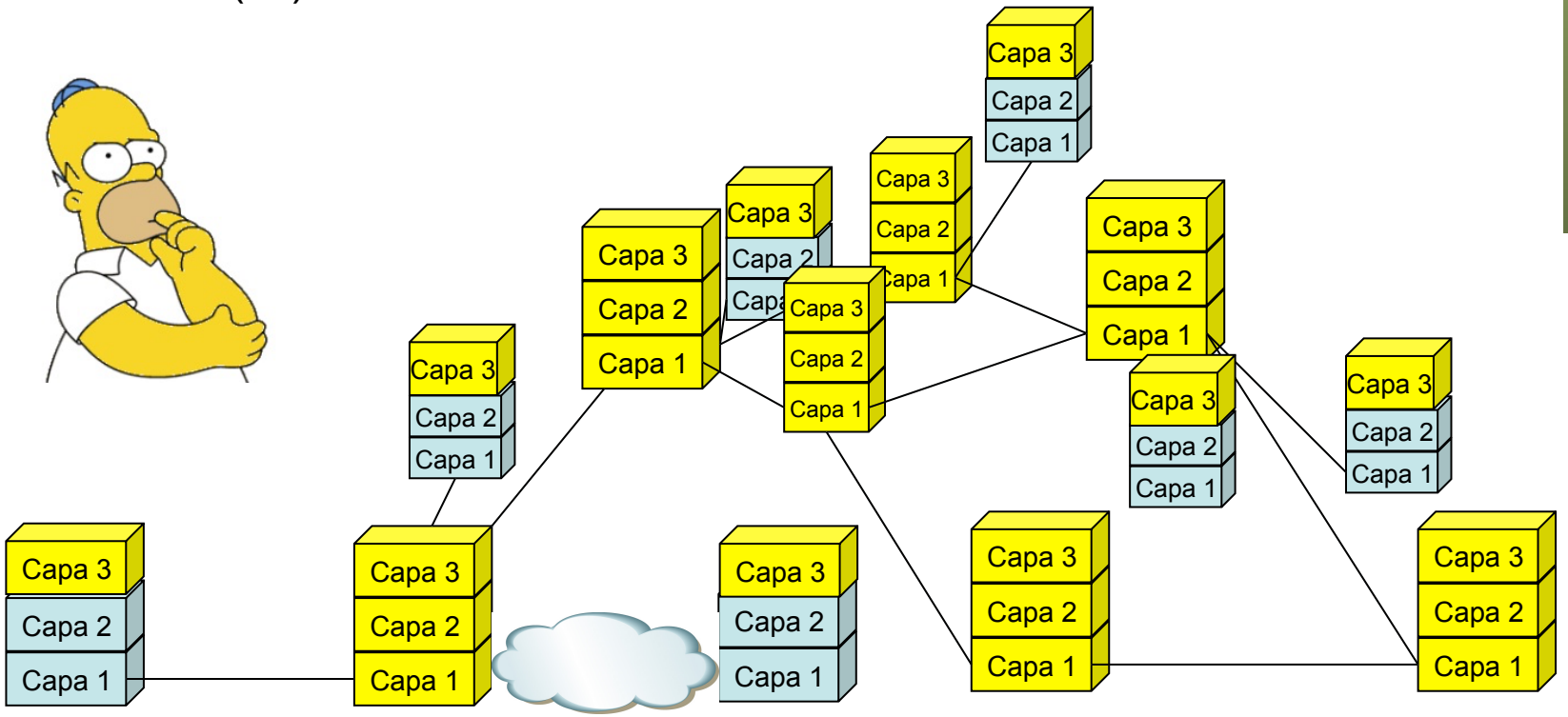


Ethernet vs Layer 3

- ¿Y los bucles?
- Tendríamos problemas con un protocolo de encaminamiento de capa 3 pues la trama Ethernet no tiene un TTL
- Así que se diseñó (Radia, en un par de días) un protocolo para eliminar esos ciclos
- Sí, el *"Spanning Tree Protocol"*
- Radia Perlman firma más de 100 patentes, ha recibido premios como los "Lifetime Achivement award" tanto de Usenix como del ACM SIGCOMM y está en el "Internet Hall of fame" de la ISOC

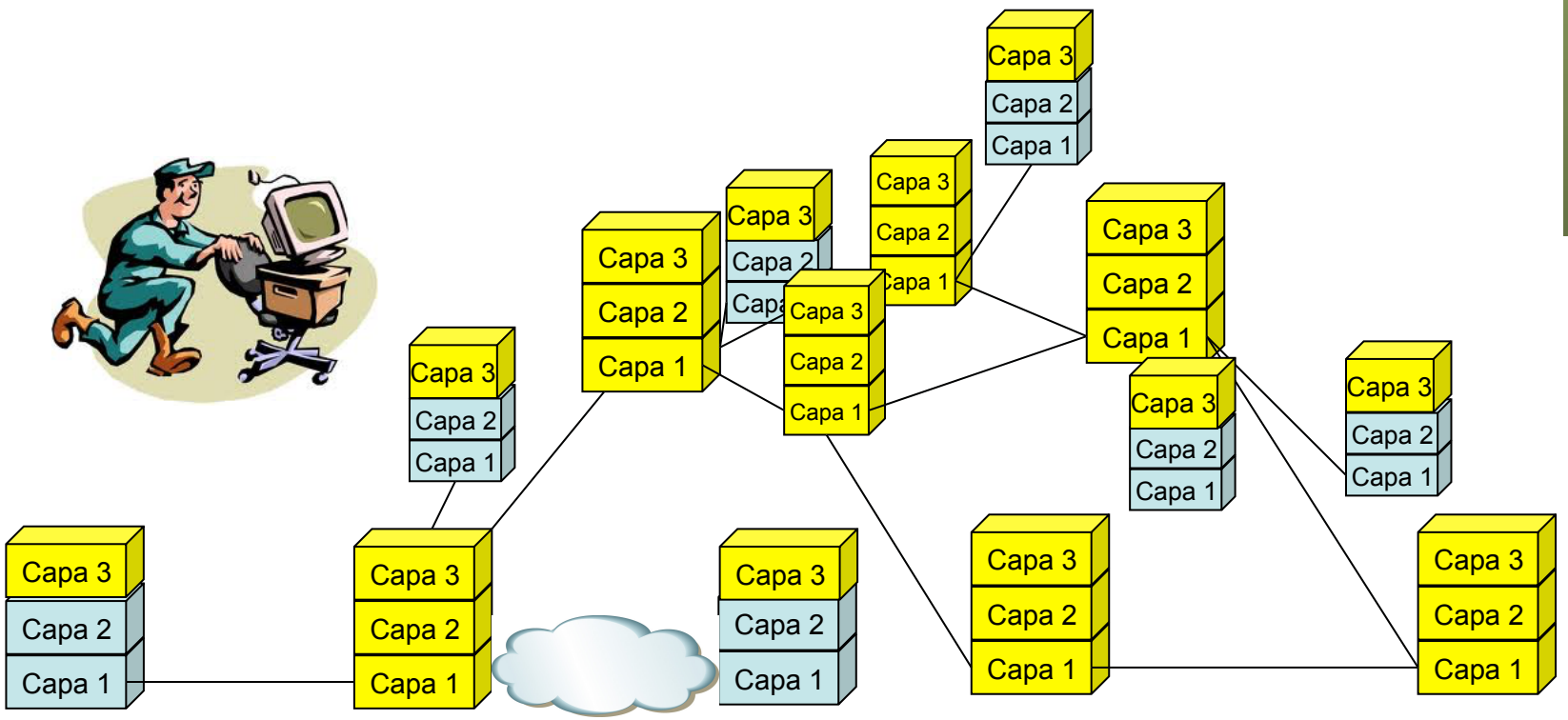
Layer 2 vs Layer 3

- ¿Quiere decir todo esto que sería mejor tener todo conmutadores en capa 3?
- Algunos enlaces serían punto-a-punto y otros serían medios compartidos (¿WiFi?)
- Ahora sí tenemos un protocolo de capa 3 implementado en todos los hosts (IPv4)
- Mmmm (…)



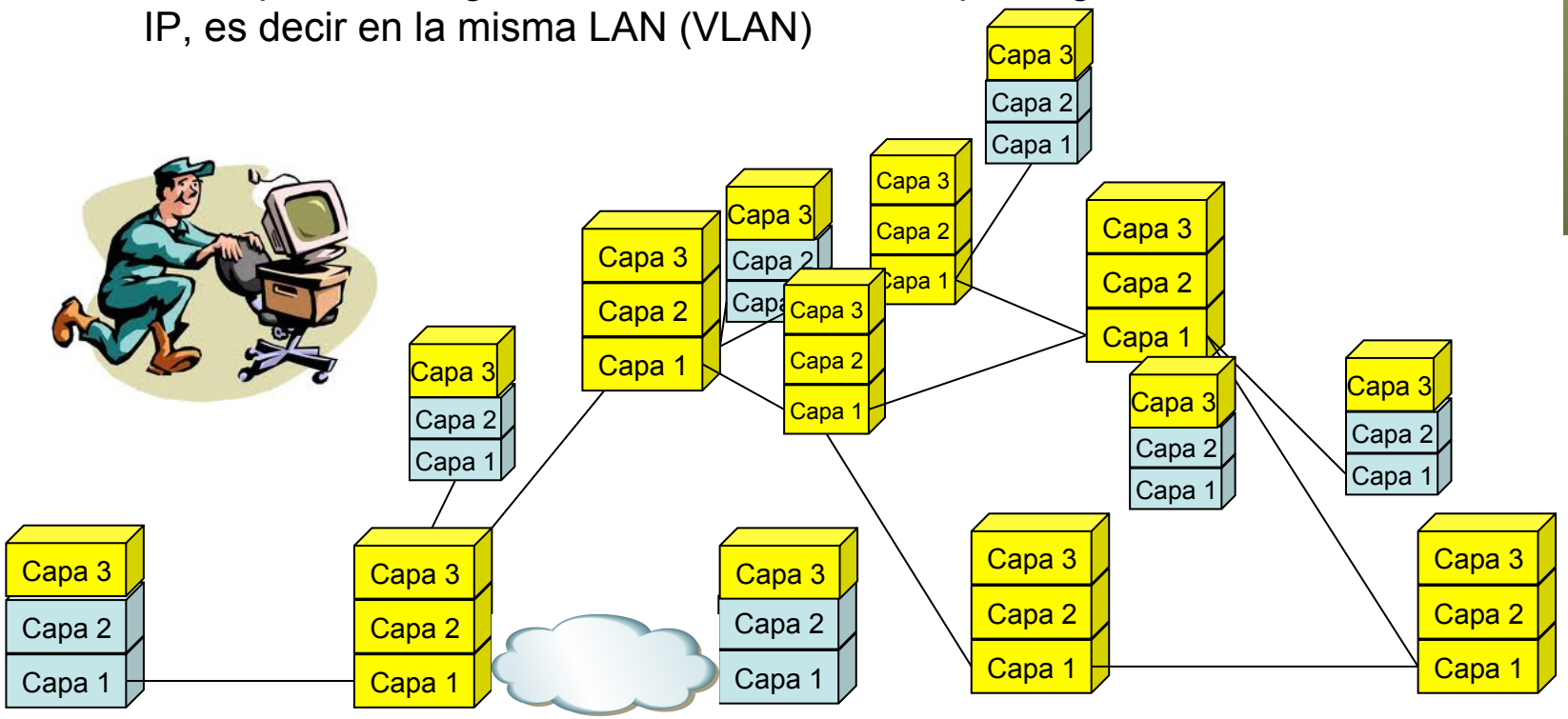
Layer 2 vs Layer 3

- Con IPv4 tenemos un problema:
  - Cada enlace es una subred
  - Si un host cambia de enlace cambiaría de dirección
  - Es decir, no podemos mover un host y mantener su dirección de capa 3
- ¿Pero queremos mover hosts frecuentemente? (…)



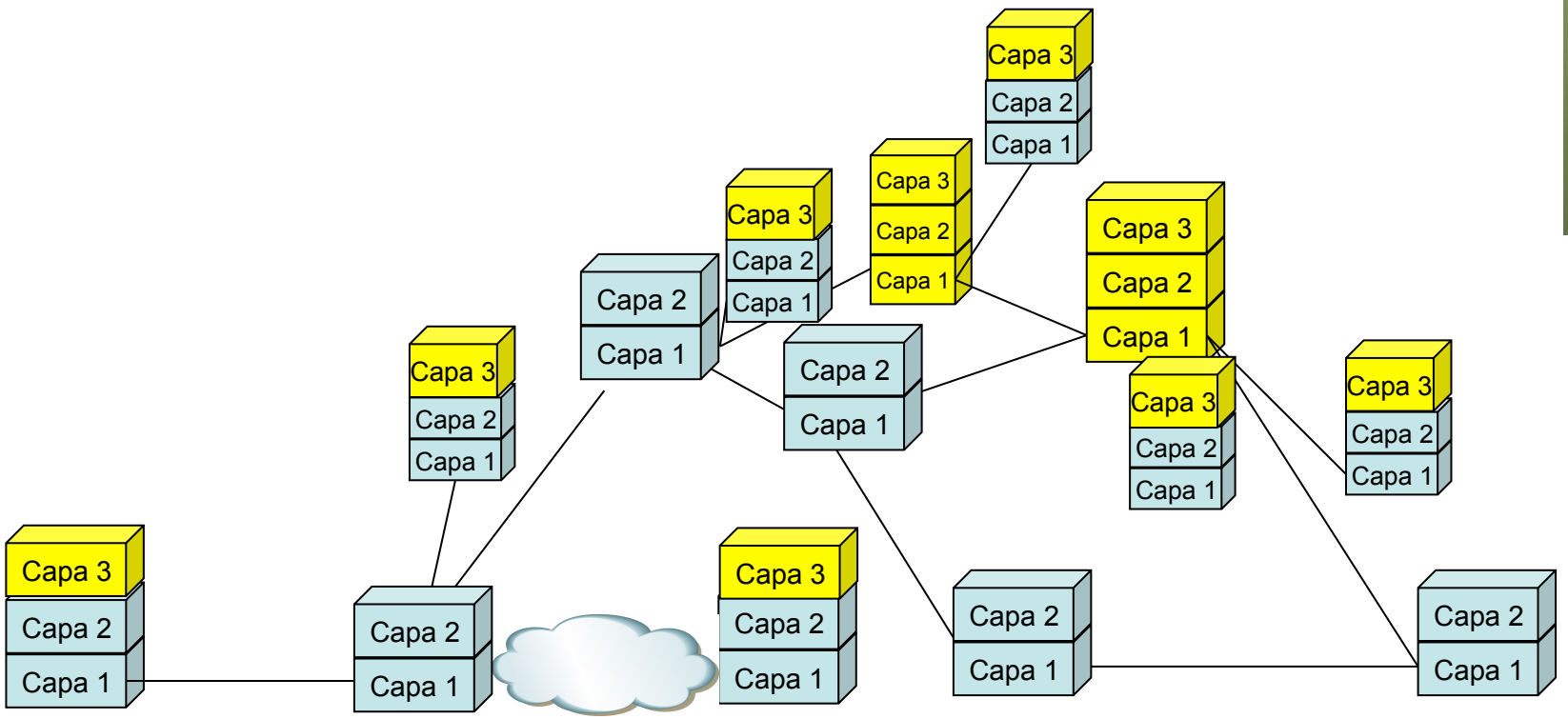
Layer 2 vs Layer 3

- Con IPv4 tenemos un problema:
  - Cada enlace es una subred
  - Si un host cambia de enlace cambiaría de dirección
  - Es decir, no podemos mover un host y mantener su dirección de capa 3
- ¿Pero queremos mover hosts frecuentemente?
  - Veremos que movemos máquinas virtuales
  - Para que mantengan la dirección IP tienen que seguir en la misma subred IP, es decir en la misma LAN (VLAN)



Layer 2 vs Layer 3

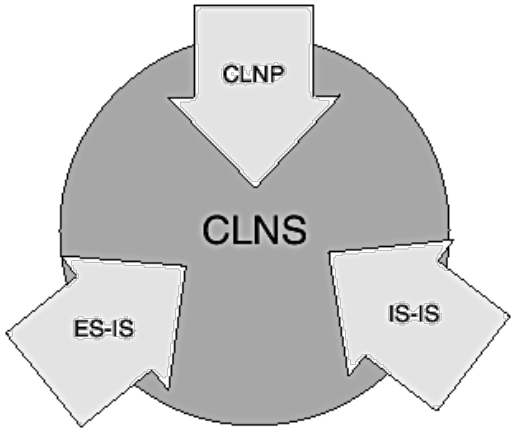
- Entonces tenemos que mantener ciertos dominios capa 2
- Pero dentro del dominio capa 2 Ethernet tendremos STP y eso nos deshabilita enlaces
- Veremos soluciones para mejorar la LAN Ethernet sin emplear STP



Layer 3 wars

Layer 3 vs Layer 3

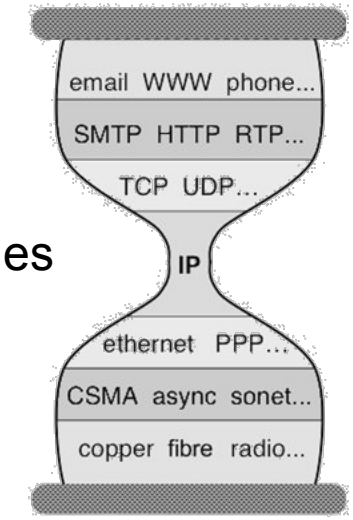
- Otro ejemplo
- ¿Sabéis qué es CLNP?
- *ConnectionLess Network Protocol*
- Protocolo de capa 3 de la pila OSI orientado a datagramas
- O sea, como IP pero en la familia OSI
- ¿Sabéis de qué tamaño son sus direcciones?
- 20 bytes (longitud variable)
- Las de IPv4 son de 4 bytes, las de IPv6 son de 16 bytes
- ¿Y qué tiene de malo?
- En su día (en los 90s) de hecho era mejor que IPv6 (mejores mecanismos de autoconfiguración, IS-IS, etc)
- ¿Por qué no se cambió a IPv6 entonces?
- ¿Y para qué se iba a cambiar?



Principios

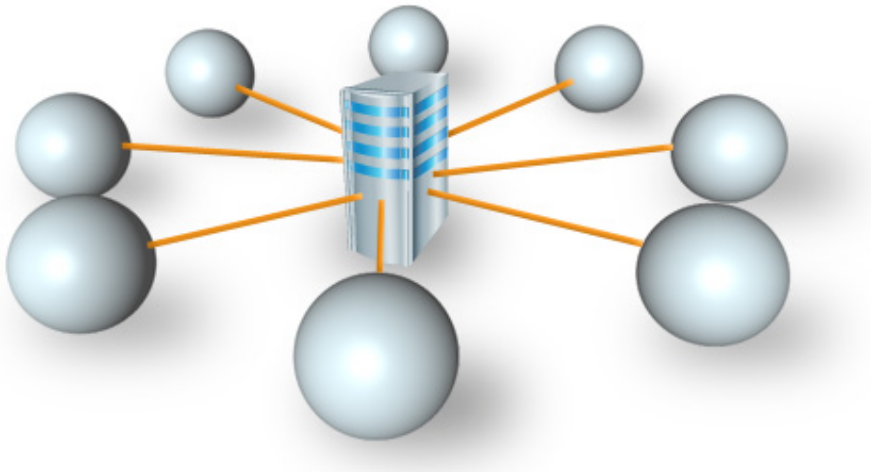
End-to-End principle

- [RFC3439] *"end-to-end protocol design should not rely on the maintenance of state (i.e., information about the state of the end-to-end communication) inside the network. Such state should be maintained only in the end points, in such a way that the state can only be destroyed when the end point itself breaks."*
- No quiere decir que no haya estado en la red, que lo hay (por ejemplo las tablas de rutas)
- Quiere decir que no interactúa directamente con los protocolos en los hosts
- Esto busca hacer simple la red
- La complejidad impide que escale
- La complejidad lleva a mayor CAPEX y OPEX
- Pero no penséis que los routers de hoy en día son simples



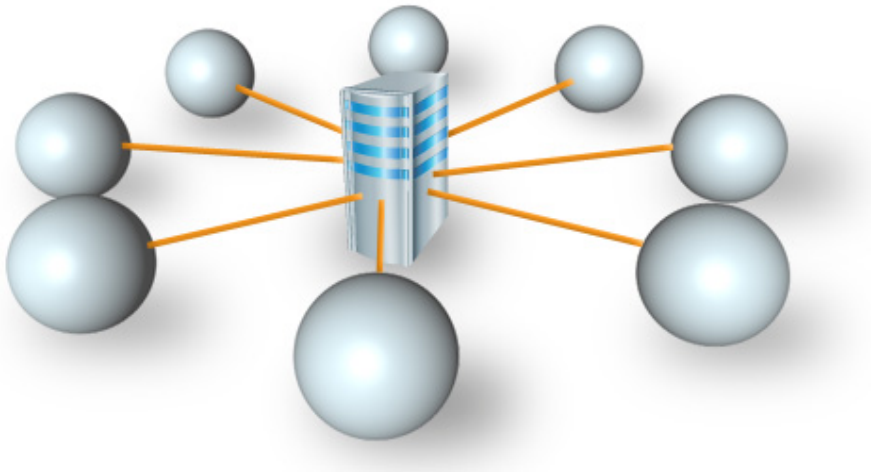
Internet architecture

- El objetivo que tenían era que si los hosts de los extremos no se colgaban debían poderse comunicar si los routers funcionaban
- Es decir, no depender de otros servidores que controlaran la red, como sucedía en la red telefónica
- El problema es que no se sabía cómo construir algoritmos distribuidos que permitieran recuperar la red ante fallos en enlaces
- Los algoritmos distribuidos no son sencillos
- En los últimos años hemos aprendido bastante sobre ellos
- Muchas veces “por las malas”



Internet architecture

- ¿Una arquitectura con elementos de control centralizados es mala?
- Es más parecido al control en la red telefónica
- Tienes puntos críticos de fallo…
- …pero puedes tener replicada su funcionalidad
- También son cuellos de botella…
- …pero puedes repartir la carga entre varios
- Va en contra de los principios de Internet…
- …pero es el fundamento de las SDNs (Software Defined Networks)



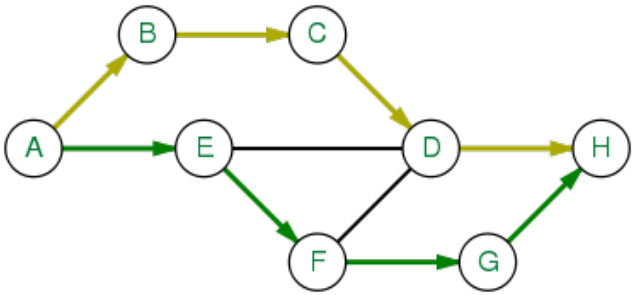
OpenFlow

Internet architecture

- ¿Una arquitectura con elementos de control centralizados es mala?
- Es más parecido al control en la red telefónica
- ¿Es tan mala la red telefónica?
  - La voz funciona bastante bien
  - Lleva décadas dando QoS extremo a extremo
  - Sin necesitar gran sobredimensionamiento
  - Ni la complejidad de los mecanismos de QoS en la red de conmutación de paquetes
  - ¿Quién se dejaría operar remotamente donde el médico controla el robot a través de la Internet?
  - Pensemos en un parámetro básico: mantener la conectividad (no hablemos de throughput o delay) (…)

Internet architecture

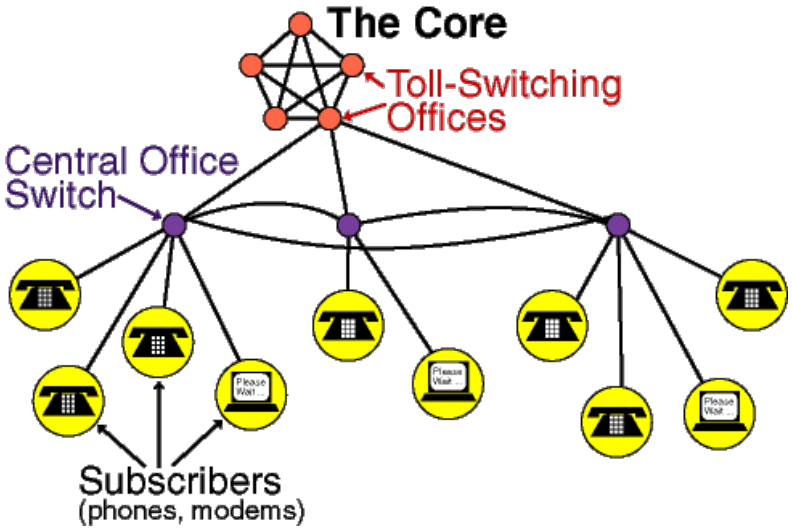
- Mantener la conectividad ante fallos en equipos y en enlaces
- Necesitamos calcular caminos alternativos rápidamente
- Con algoritmos distribuidos
- Para sobrevivir ante fallos deben ser caminos disjuntos



- IP dentro de LSPs MPLS, sobre wavelengths, dentro de fibras, dentro de grupos de fibras
- ¿Pero cómo sabe el nivel IP si dos caminos que ha calculado son físicamente disjuntos?
- ¿Y si van por la misma fibra? ¿O las fibras por el mismo tubo? ¿O los tubos por la misma canalización?
- Por ejemplo se hunde un túnel por donde pasan muchas canalizaciones

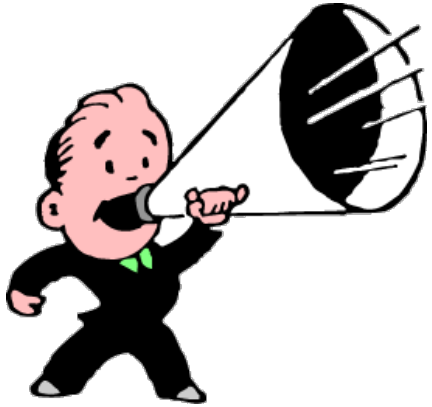
Internet architecture

- La red telefónica juega con ventaja
- Normalmente desde la fibra entre centrales hasta el teléfono final era controlado por la misma operadora
- También su tráfico es muy predecible
- No solo por las fuentes y destinos involucrados sino por la cantidad de tráfico en cada flujo
- Eso permite calcular la capacidad necesaria en los enlaces
- Y permite decidir caminos óptimos en la red
- Incluso se pueden adaptar a patrones horarios (se pre-calculan rutas para franjas horarias)



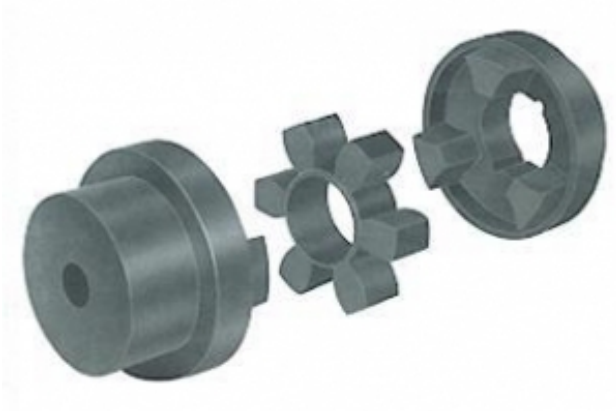
Amplification principle

- [RFC3439] *"…there are non-linearities that occur at large scale which do not occur at small to medium scale"*
- En redes grandes incuso sucesos pequeños pueden tener grandes consecuencias
- Es decir, pequeñas perturbaciones pueden desestabilizar el sistema
- Ejemplo: añadir un pequeño número de enlaces puede hacer la resolución del routing mucho más compleja
- Ejemplo: descartar una celda ATM lleva a perder un paquete completo
- Para evitarlo se intenta que los cambios locales tengan efectos locales, no globales



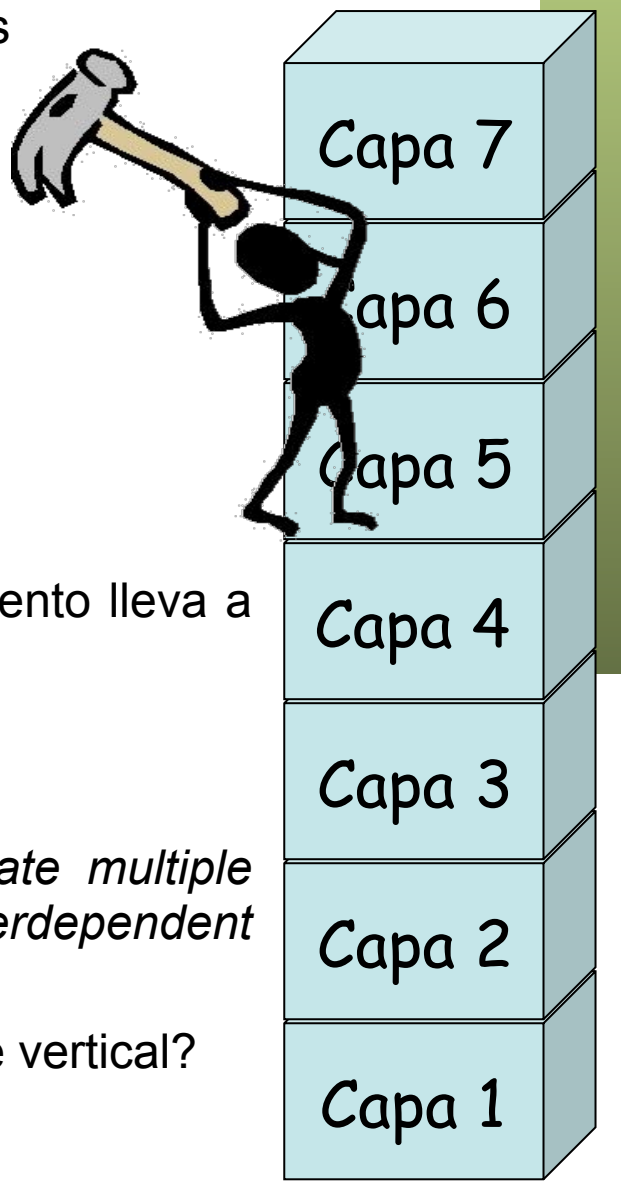
Coupling principle

- [RFC3439] *"…as things get larger, they often exhibit increased interdependence between components."*
- Acoplamiento horizontal: en la misma capa de protocolos
- Acopamiento vertical: entre capas
- Ejemplo: conexiones TCP sincronizan el comportamiento de su ventana de control de congestión al compartir cuello de botella
- Una forma de reducir el acoplamiento es introducir aleatoriedad



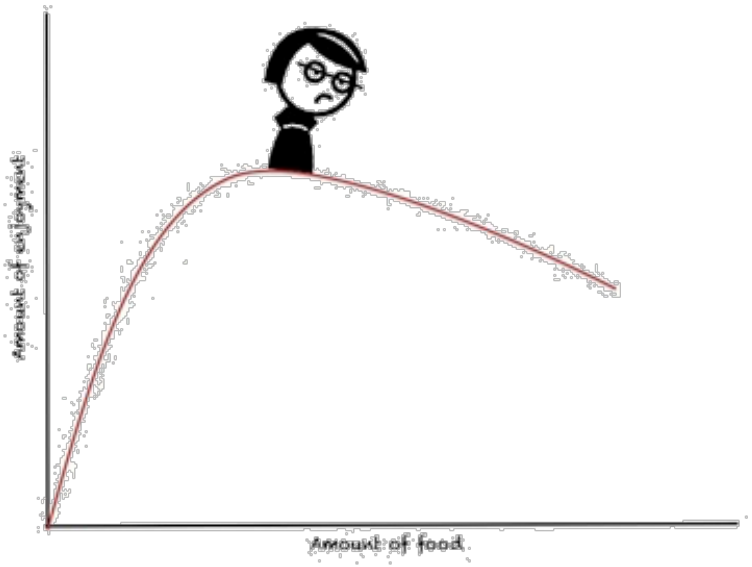
Arquitectura de capas

- La integración vertical nos permite repartir tareas
- Permite aislar implementaciones
- Las capas implementan funcionalidades como:
  - Control de errores
  - Control de flujo
  - Fragmentación
  - Multiplexación
  - Control de conexión
  - Direccionamiento
- Sin embargo la misma independencia y aislamiento lleva a que se reimplementen las funcionalidades
- Puede llevar a mayor complejidad
- Y acabamos violando el principio de simplicidad
- [RFC 1925] *"It is always possible to agglutinate multiple separate problems into a single complex interdependent solution. In most cases this is a bad idea."*
- ¿Tal vez es mejor una separación horizontal que vertical?



La optimización es dañina

- Un poco de optimización está bien
- Pero la optimización pasado cierto punto introduce complejidad y mayor acoplamiento entre las capas
- Esto lleva a sistemas menos fiables
- Ley de los rendimientos decrecientes (*Diminishing Returns*): al aumentar algo que da beneficio, cada vez incrementa en menos el beneficio, hasta llegar a poder reducirse



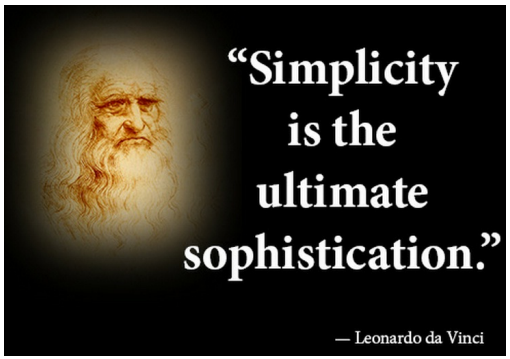
Packet- vs Circuit- Switching

- ¿Qué es más eficiente?
- La conmutación de paquetes permite mayor eficiencia mediante la multiplexación estadística
- Sin embargo la utilización en los enlaces de conmutación de paquetes es muy baja
- ¿Por qué?
  - Es muy difícil predecir el tráfico así que se suele sobredimensionar grandemente la capacidad
  - Si falla otro enlace puede redirigirse todo su tráfico por éste, así que mejor dimensionemos con capacidad disponible para eso
  - es decir, es como si hiciéramos un 1:1
  - Las tecnologías tienen escasa granularidad (10GE pasa a 100GE)
  - Conseguimos QoS mediante *over-provisioning*

Packet- vs Circuit- Switching

- ¿Qué es más **simple**?
- El principio end-to-end le da simplicidad a Internet, ¿sí?
- IP es simple, pero los routers y los protocolos (algoritmos distribuidos) no lo son
- Software es más complejo en routers
- Las operaciones que deben hacer por paquete son más complejas que las que hace un conmutador de circuitos
- El hardware del router es más complejo
- El router consume más potencia por esto último
- El control es más simple en CC (estático, out-of-band)
- ¿QoS en Internet? CC la tiene desde su concepción

- ¿Qué es más simple?
- ¿Qué es más eficiente?

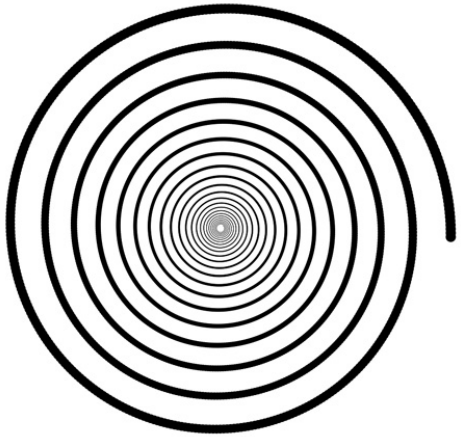


Debugging

- Requiere experiencia con gran cantidad de componentes, tecnologías y protocolos
- Muchas veces requiere visitar la configuración de los equipos uno a uno
- Las herramientas son primitivas
- El problema puede no estar en la red sino en algún servicio auxiliar (¿DNS?)

99.999%

- ¿Podemos mejorar la tecnología para lograr redes fiables durante el 99.999% del tiempo?
- La gran mayoría de los fallos están causados por humanos o malos procedimientos
- Es decir, mejorando la tecnología tal vez logramos eliminar el … ¿20% de los fallos?
- ¿Y cómo lo hacemos? Aumentando la complejidad del sistema
- Mayor redundancia, caminos alternativos, algoritmos distribuidos para recuperarse ante fallos, etc
- Lo cual hace más frágil el sistema porque hay más puntos en los que los humanos nos podemos equivocar
- Es decir, reducimos los fallos de la tecnología pero aumentamos los fallos de los humanos



Moraleja

RFC 3439

Moralejas

- Hay mucha herencia e historia
- Las soluciones actuales no tienen por qué ser técnicamente las mejores
- Pueden estar condicionadas
  - Relaciones de poder entre empresas en su momento
  - Visiones subjetivas (“mi protocolo es mejor”)
  - Cuestiones económicas (“no puedo tirar toda la red que tengo para poner esa nueva tecnología”, “esto es ahora más barato”)
  - Desconocimiento de otra tecnología por la gente que defiende una
  - Relaciones entre grupos de “estandarización”
- Al aumentar la escala de la red hablamos de sistemas complejos
- Para hacerlo más fiable lo hacemos más complejo
- A más complejo, más fallos humanos
- A más fallos humanos más frágil