Preguntas de Final de Redes (TC)

Teoría de la información

1. ¿Qué es información?

Es una medida de la probabilidad de ocurrencia de un evento. Cuanto menos probable es un evento, más información aporta. Se calcula como $I(e) = \log(\frac{1}{P(e)})$ (para base 2 se mide en *bits* de información)

2. ¿Qué es la *entropía*?

Dada una fuente de información, con sucesos $S_1,..,S_n$ la *entropía* es la **cantidad media de información** (esperanza) de la misma: $H(S) = \sum_{S1..Sn} P(S_i) \times I(S_i)$. Es por tanto también una **medida de incertidumbre,** cuanto mayor es la entropía, mayor es la incertidumbre sobre los datos de la fuente**.**

3. ¿Qué es una fuente de información? Describir tipos de fuentes.

Es el productor de símbolos *Si*, pertenecientes a un alfabeto finito, a ser procesados. De acuerdo a dependencia probabilística entre cada símbolo y los subsiguientes, se caracterizan diferentes tipos de fuente:

- Fuente de *Memoria Nula*: todos los símbolos son **independientes** entre si.
- Fuente *N-aria*: su alfabeto consta de N símbolos
- Fuente de *Markov* de orden *n*: la probabilidad del carácter S_{n+1} depende de los *n* caracteres precedentes.

4. ¿Cómo se minimiza/maximiza la entropía? ¿Qué pasa si varío la forma de medirla?

La entropía depende directamente de las probabilidades de cada símbolo. Luego H(S) es **máxima** cuando los símbolos de S son *equiprobables* y se reduce cuando la ocurrencia de un símbolo aporta información sobre la probabilidad de ocurrencia de otro. Es **mínima** cuando existe un símbolo *Si* con $P(S_i)=1$ y el resto es 0.
Como propiedades de la codificación de información dependen del valor de la entropía (inecuación de Kraft), si se mide de otra forma puede alterarse esta ley.

5. Cuando se realiza una *extensión* de una fuente de información, la entropía de la nueva fuente es H(s^n) = n *H(S). Explique con un ejemplo el concepto de extensión de una fuente.

La extensión de orden *N* de una fuente consiste en formar un nuevo alfabeto, agrupando *N* símbolos del alfabeto original por palabra (particiones del conjunto. Ej de orden 2: {1,2} -> {11,22,12,21})

6. ¿Qué es un *código* y cómo se caracteriza?

Es un proceso orientado a lograr una representación más eficiente, en términos de caracteres utilizados, de una fuente de información. Establece una correspondencia entre símbolos de una fuente y símbolos del alfabeto de un código.
Un código se caracteriza como:

- *no singular*: todas sus palabras son diferentes
- *unívoco*: si su extensión de orden *n* es no-singular para cualquier valor finito *n*.
- *instantáneo:* las palabras pueden ser decodificadas sin conocer los símbolos precedentes (condición: no existe palabra que sea prefijo de otra).

7. ¿Qué es la longitud media de un código? ¿Cómo se relaciona con la entropía?

Es el valor promedio de la longitud de una palabra codificada. Sea L_i la longitud del símbolo S_i codificado, entonces se calcula como: $L = \sum_{1...K} P(S_i) \times L_i$
Si se exige que el código sea instantáneo, para que sea decodificable, entonces, por la **inecuación de Kraft** debe cumplirse que la **entropía sea menor o igual que la longitud media**: $H(S) \leq L$
Entonces **la mínima longitud de un código** se consigue cuando la longitud de cada palabra codificada es igual a la **información** de la misma: $l(s) = \log(\frac{1}{P(s)})$ (código *óptimo*)

8. Enuncie dos conclusiones prácticas de la teoría de la información y la codificación

- La entropía es máxima cuando los símbolos son equiprobables
- La longitud de un código, para se decodificable, no puede ser menor a la entropía.

9. ¿Cómo aplicaría un mecanismo compresión en un texto ASCII basándose en estadística?

Se asignan más bits a los símbolos menos probables y viceversa (código de Huffman). Este código es óptimo.

Capa 1: Física

10. ¿Qué es el *ancho de banda* en qué afecta la capacidad de transferencia?

Es el **rango de frecuencias** en el que puede transmitirse una señal **analógica**. Depende del medio físico utilizado. Restringe la máxima *capacidad /velocidad* de transferencia de datos según el teorema de Shannon.

11. Enunciar y describir el *Teorema de Shannon*

El teorema define una limitante para la capacidad máxima de transferencia de un canal físico, dada su propiedad de relación *Señal-Ruido* (Signal-to-Noise ratio).

El teorema define $C = BW(hz) \times \log_2(1 + SNR)$ tal que $SNR = 10 \cdot \log(\frac{S}{R})$

12. Enunciar y explicar el *Teorema de Nyquist*

Si una señal fue pasada por un filtro *pasabajos* de ancho de banda *BW*, puede reconstruirse completamente tomando 2*BW muestras por segundo. Ej: canal de voz transmite a 4KHz, luego se muestrea a 8000 muestras por segundo.

13. Para transmitir voz por canal telefónico se utilizan 8 bits mientras que para reproducir un CD se usan 16 bits ¿por qué sucede esto?

Porque, por teorema de N., para reconstruir una señal de mayor "calidad" (ancho de banda) se requiere el doble de muestras por segundo.

14. Describir métodos de control de errores en capa 1

¿Mitigación de errores? (amplificadores, ecualizadores, etc) ¿codificación de señales? (mancheste, 4b/5b, etc)
La codificación de señales permite

15. ¿Cómo es el proceso de conversión de las señales analógicas a digitales?

Este proceso se realiza mediante un ***codec***, el cual muestrea al doble del ancho de banda obteniendo un *tren de pulsos de amplitud variable* y determina el valor en N bits de la señal digital.
Los tipos de modulación son: PCM, Delta, etc.

16. ¿Para qué sirve un MODEM? ¿En qué se diferencia de un *conversor analógico-digital*?

El **MODEM** realiza una conversión de **una señal digital a una analógica** y viceversa para la transmisión de la misma a través de un canal. La conversión *digital*-analógica, denominada **modulación**, se realiza según las características intrínsecas de cada medio (ruido, atenuación, velocidad,...) mediante la variación de frecuencias amplitudes y fases de la señal analógica resultante.
Se diferencia de un *codec* en el sentido que la transformación analógica-digital toma una señal ya modulada de un canal de transmisión, mientras que el codec la toma de una entrada analógica "real". La conversión digital-analógica es para transmisión, mientras que en un *decoder* es para reproducción.

17. ¿Cómo se multiplexan las señales en un canal troncal?

Las dos técnicas principales son: **TDM** y **FDM** (división por tiempo y por frecuencia). La primera asigna todo el BW a una transmisión durante un tiempo dado (round-robin). La segunda divide las comunicaciones paralelamente en el rango de frecuencias disponible (BW). **TDM** puede utilizarse con circuitos completamente digitales, y se utiliza únicamente para señales digitales, mientras que FDM requiere circuitería analógica.

18. ¿Qué es el troughput? ¿Cuál es el limitante de esta medida?

Tiene en principio dos interpretaciones: *Bandwith* (cantidad de datos que el canal permite enviar por unidad de tiempo) y *end-to-end* (cantidad de datos enviados que llegan efectivamente al receptor por unidad de tiempo).
El limitante es siempre en última instancia el ancho de banda.

19. ¿Qué tipos de onda se pueden pasar por un canal de trasmisión (cuadradas, senoidales,...)?

Las únicas ondas que pueden transmitirse en un canal, son las senoidales. Las ondas cuadradas no existen fuera del marco teórico, porque, al ser composiciones de diferentes armónicas, según la serie de Fourier, se necesitaría un ancho de banda infinito para formar una onda cuadrada "real". Las ondas cuadradas en la práctica son producidas con un conjunto finito de armónicos y según el nivel de señal detectado, se determina si corresponde a un 0 o a un 1.

20. ¿Por qué no puede transmitirse una onda cuadrada en un BW de 3000hz?

Las ondas cuadradas no existen fuera del marco teórico, porque, al ser composiciones de diferentes armónicas, según la serie de Fourier, se necesitaría un ancho de banda infinito para formar una onda cuadrada "real".

21. ¿Qué relación hay entre la información de la señal y el ancho de banda?

La cantidad de información que puede transmitirse en un canal en un momento dado, está limitada por el BW y la latencia del canal.

22. Si te tiene una onda cuadrada de 10Khz y se inyecta en un canal con BW de 15Khz ¿cómo es la señal que llega al receptor?

La señal primero tiene que ser modulada mediante un MODEM, generando la señal senoidal correspondiente. En destino se efectúa la operación inversa, tomando la señal analógica y generando la cuadrada correspondiente, mediante series de Fourier.

23. Describa las técnicas de Spread Spectrum para transmisión de señales y sus implementaciones: Direct Sequence y Frecuence Hoping.

Ambas son **técnicas de transmisión de señales** electromagnéticas por medios inalámbricos (radio, radar, etc). ***Spread Spectrum*** aprovecha la existencia de rangos de frecuencia no licenciados para distribuir la señal en un abanico de frecuencias más amplio que el necesario para transmitir. Distribuye la señal de manera de evitar concentrar la potencia en una sola frecuencia, utilizando frecuencias ya "ocupadas" sin interferir. Se implementa de dos formas:

Direct Sequence: Each bit of data is represented by multiple bits in the transmitted signal so that, if some of the transmitted bits are damaged by interference, there is usually enough redundancy to recover the original bit. For each bit the sender wants to transmit, it actually sends the exclusive-OR of that bit and n random bits. As with frequency hopping, the sequence of random bits is generated by a pseudorandom number generator known to both the sender and the receiver. The transmitted values, known as an n-bit chipping code, spread the signal across a frequency band that is n times wider than the frame would have otherwise required..

Frecuence Hoping: Mediante un selector pseudo-aleatorio de frecuencias, sincronizado entre emisor y receptor, se transmite la señal, saltando de una frecuencia a otra. Para un receptor que no tenga sincronizado el generador, esta señal se ve como "ruido".

Capa 2: Enlace (Link)

24. ¿Qué es una red de computadoras?

Es un conjunto de computadoras conectadas a través de un medio físico mediante un protocolo común. Con la finalidad de ofrecer servicios y compartir recursos.

25. Describir tipos de redes: Ethernet (802.3), Wireless (802.11), Token Ring (802.5)....

Ethernet 802.3

- conecta los equipos mediante dispositivos de enlace que incorporan la capa MAC (medio físico) y LLC (independiente de la capa subyacente).
- El medio de transmisión es el cable coaxial o UTP de 10Mbps. La máxima distancia entre equipos es de 2500m, amplificando la señal mediante hubs, en la cual un frame tarda 56 microseg. en llegar de una punta a la otra.
- Implementa el protocolo CSMA/CD (Carrier Sense Multiple Access / Collission Detection): En esta red, todos los equipos tienen acceso simultáneo al medio, sensando continuamente si está en uso, y una vez liberado el canal pueden transmitir cuando lo deseen.
- Si dos equipos transmiten al mismo tiempo, ocurre una *colisión*.
- Para evitar que una estación transmita mientras lo hace otra, cada frame debe ocupar el canal por completo (los 2500m) por el tiempo que tarda en ir y volver la señal (por si otro transmite un 'delta t' antes de que llegue el frame) , por lo que, en base a BW * delay, se debe transmitir frames de 64bytes.
- En el caso de una colisión, las estaciones deberán volver a transmitir. Para reducir el riesgo de otra colisión, se implementa el algoritmo de *Exponential Backoff*: cuando el canal se libera los equipos no transmiten inmediatamente sino esperan un tiempo aleatorio entre 0 y t_{max} para transmitir. Luego de una colisión, por cada reintento se aumenta t_{max} en $2^{\#\ intentos}$.

Token Ring 802.5

- Presenta una **conexión en "anillo"** entre los nodos. Cada nodo tiene su par *upstream* y *downstream*. Todos tienen acceso al medio, pero solo pueden **transmitir por turnos.**
- Se propaga una señal de *token* por la red que todos los nodos toman y reenvían. Cuando un nodo quiere transmitir, espera a recibir el token, lo saca de la red y transmite durante un tiempo THT. Esta transmisión se propaga por todos los nodos hasta retornar al origen. Luego reinserta el token.
- Si pocos nodos transmiten a la vez, cuando mayor sea el THT mejor la eficiencia de la red y viceversa.
- Por esta razón tiene un **comportamiento determinístico** en cuanto a la transmisión, no como 802.3.

Wireless 802.11

- Sistema de Red de transmisión **inalámbrica** (radio/infrarojo). Satisface *movilidad* y *ad-hoc networking*.
- Transmite mediante FHSS, DSSS, etc. (ver pregunta 23)
- Debido a los problemas de *Nodo Oculto* y *Terminal Expuesta* (ver 27) el protocolo **CSMA/CD NO sirve para Wireless**. Se implementa **MACA/MACAW** (Multiple Access Colision Avoidance).
    - El emisor envía un Request To Send con la longitud de los datos a transmitir
    - El receptor contesta con un Clear To Send con la misma info y habilita la transmisión.
    - Cualquier estación cerca del emisor esucha el RTS y espera. Lo mismo con las estaciones cerca del receptor que escuchan el CTS.
- MACAW incorpora ACKs para cada frame, detección de portadora CSMA/CA y exponential backoff.

26. ¿Puede garantizarse el acceso al medio en tiempo acotado en WANs? ¿Con qué protocolos?

Ethernet, al no ser determinístico, NO puede garantizar acceso al medio para transmitir en tiempo acotado. Token Ring SI puede, siendo el tiempo máximo de espera una función del THT, la cantidad de nodos y el tiempo de transmisión del token en la red (TRT).

27. Describir los problemas de *Nodo Oculto* y *Terminal Expuesta* de 802.11

***Nodo Oculto**:* ocurre cuando dos terminales A y C, fuera de rango, quieren transmitir a B y, al no detectar la señal una de otra, transmiten y colisionan en B.

Terminal Expuesta: sucede cuando una terminal B, en rango de dos terminales A y C, transmite a A y por lo tanto la señal es sensada por C, quedando ésta bloqueada de transmitir a otro posible nodo D.

28. ¿Cómo se transmiten frames en un canal Ethernet?

Para poder dividir la cadena de bits en frames, se utilizan varias técnicas**:** *Conteo de Caracteres, Flagging con Byte Stuffing* y *Flagging con Bit Stuffing*.

- **Conteo de caracteres**: se envía en un header la cantidad de caracteres de frame. Muy susceptible a errores debido a modificaciones de la señal
- **Flagging con Byte Stuffing**: separa frames mediante un byte de flag. Si el flag aparece en la cadena transmitida, lo escapa mediante un byte especial ESC. Está atado a usar caracteres de 8bits (Unicode usa 16).
- **Flagging con Bit Stuffing**: permite transmitir cualquier número de bits por frame y cualquier número de bits por carácter. Cada frame empieza y termina con una secuencia de bits 0111110. Si el emisor detecta una cadena de 5 1's, inserta un 0 antes del último.

29. ¿Qué control de errores se realiza en este nivel? Diferencias con nivel 4

Los principales controles de errores a nivel de enlace son: *Bit de Paridad* y *Cyclic Redundancy Check* (CRC):

Bit de Paridad: consiste en agregar a los bits del frame una cantidad *r* de bits de paridad, buscando que la cantidad total de 1's sea par (o impar). Con un frame 100100 y un bit de paridad impar, se completaría la *palabra clave* (codeword: frame + paridad) con un 1 = 1001001. Un bit de paridad implica una ***distancia de Hamming** de* 2, dado que 2 errores en bits únicos genera otro código válido.
En protocolos Wireless con gran tasa de errores, se utilizan bits de paridad para matrices de frames, donde la paridad actua por filas y columnas de bits.

Cyclic Redundancy Check (CRC): consiste en tomar los frames de *k* bits como polinomios de bits de orden k-1, donde el más significativo corresponde a x^{k-1}. Luego ambas partes acuerdan un polinomio generador G(x) y para cada frame generan un polinomio M(x) divisible por G(x) (utilizando división módulo 2 y restando el resto al divisor). Luego el receptor al recibir un frame calcula el M(x) y verifica que el resto de la división por G(x) sea 0, sino es porque ocurrió un error.

La diferencia con controles de errores de niveles superiores es que **los de capa de enlace son más complejos y robustos** y generalmente implementados por *hard* mientras que los de capa 4 son muy fácilmente implementables por *soft* y rápidos de calcular. En capa 4 se utiliza simplemente un *Checksum* del frame, sumando cada palabra de 16 bits del header en complemento a 1, el cual detecta errores simples que pueden no haberse detectado a nivel 2.

30. ¿Cómo se implementa QoS en ATM?

ATM provee 5 servicios básicos para proveer QoS: Variable Bit Rate (VBR), Constant Bit Rate (CBR), Available Bit Rate (ABR), Unespecified Bit Rate (UBR) y Cell Loss Priority (CLP). Los 4 primeros controlan la variación en los tiempos de transmission (*jitter*)

VBR: para comunicaciones en tiempo real (audio, video), permite definir el **máximo delay** en la recepción de un paquete. Una aplicación receptora puede definir la tasa de reproducción de un video en base a esta variable, sabiendo que ningún paquete llegará más tarde que ese valor. En este caso se tiene un promedio y un pico máximo, que se definen al momento de crear la conexión.

CBR: Tasa de demora **constante** de envío de paquetes. Caso particular del anterior, donde la tasa promedio y el máximo delay son iguales. Muy usado en comunicaciones telefónicas con BW acotado y de fácil implementación.

UBR: transmisión *best effort* sin tiempos determinados de entrega (NO APLICA).

ABR: mecanismo que opera sobre circuitos virtuales de detección de congestión de la red y definición de la tasa de transmisión adecuada. Para esto se intercambian *celdas* (paquetes ATM) especiales llamadas Resource Managent. El emisor envía estas celdas para obtener datos de congestión de la red y determinar el BR (Mecanismo de control de congestión de circuito cerrado).

CLP: mecanismo para determinar la prioridad de los paquetes/celdas de ATM. Existe en el header un bit CLP que indica la probabilidad de un paquete de ser descartado en caso de congestión. Si CLP=0, el paquete està asegurado (insured) y no se descarta. Si vale 1, entonces el paquete es de tipo Best-Effort y puede descartarse en casos de congestión.

31. ¿Porqué se dice que Ethernet es *no-determinístico* y Token Ring *determinístico*?

En Ethernet, cuando en enlace se encuentra liberado, cualquier nodo puede transmitir en cualquier momento (o con probabilidad *p*) y no hay prioridades entre nodos. En Token Ring, un nodo solo transmite cuando tiene el *token* y por el tiempo determinado en el *Token Holding Time* (THT). Por esto último puede determinarse qué nodo va a transmitir en un momento dado (ver pregunta 25).

32. ¿Qué ventajas aporta una conexión de red mediante *Swtiches*?

- Conexión full-duplex entre cada terminal y el switch.
- Eliminación de colisiones
- Escalabilidad, pudiendo conectar varias redes en longitudes mayores a 2500m
- Broadcast controlado mediante VLANS

33. ¿Qué problema puede surgir al conectar una red con Switches? ¿Cómo se soluciona?

Puede que una red tenga ciclos si dos switches se interconectan por dos lados diferentes.
Se soluciona generando un árbol de switches, mediante el protocolo STP, donde se define una jerarquía de switches y se bloquean las interfaces que generan ciclos.

34. Describir el algoritmo STP (*spanning tree protocol*) ¿en qué momentos se ejecuta?

Se ejecuta al iniciar una red, agregar un nuevo switch o al caerse un enlace del arbol.
(ver pag 187 Peterson)

35. ¿Qué es un circuito virtual?

Es una transmisión orientada a conexión, donde se establece un circuito de switches por el cual se transmitirán los frames, siguiendo siempre la misma ruta. Se implementa mediante tablas de VCs en los switches e identificadores de circuito VCIs para cada frame.
Dado un frame, el switch verifica el VCI y determina la interfaz de salida correspondiente al circuito en base a la tabla.

36. ¿Puede ocurrir que en un circuito virtual se entreguen paquetes desordenados? (poco importante)

...?

37. Describir el **control** de congestión en *circuitos virtuales*

El control de congestión se realiza mediante mecanismos de *circuito abierto* y *circuito cerrado.*

Para el primer caso se toman estadísticas de las transmisiones: RTT, timeouts, etc.

Para el caso de circuito cerrado, se decide al momento de crear la conexión, la reserva de recursos necesarios para la misma, de forma de nunca recibir más paquetes para encolar que el tamaño del buffer.
De igual forma también puede reservarse un determinado ancho de banda para cada conexión virtual.

Existe también el paquete **Resource Manager** de ATM (ver pregunta 30) donde se intercambia info entre ambos nodos sobre la congestión de la red.

38. ¿Cómo se interconectan dos redes mediante múltiples enlaces Ethernet (*etherchannel*)?

La tecnología etherchannel se implementa en switches Cisco y permite aumentar el ancho de banda de un enlance entre dos switches –generalmente un backbone- mediante la agrupación lógica de de múltiples enlances Ethernet físicos. De esta forma se provee un ancho de banda igual a N*BW y redundancia en caso de fallos.

39. ¿Qué ocurre con la performance de una red Ethernet si se aumenta la cantidad de máquinas? ¿y en una Token Ring?

En una red Ethernet, dado que todos los nodos pueden transmitir en cualquier momento en que se encuentra liberado el enlace, la probabilidad de colisiones al aumentar la cantidad de nodos crece, luego se degrada la performance. En Token Ring esto no sucede dado que los nodos transmiten por turnos por un tiempo determinado.

40. ¿Puede asegurarse siempre que un paquete llegó a destino? Describa el problema de los “dos ejércitos”

No hay forma de asegurar, desde el punto de vista de un emisor, que un paquete llegó efectivamente a destino si existe la posibilidad de que el mensaje se pierda o sea modificado.

El problema de los dos ejércitos plantea la imposibilidad de que ambas partes lleguen a un acuerdo sobre el “ataque” si los “mensajeros” pueden ser eliminados (paquetes perdidos) y no pueden llevar como respuesta la misma información que recibieron inicialmente (podrían ser capturados).

En un protocolo que requiere que ambas partes estén de acuerdo en la operación a realizar, siempre hay un último emisor que confirma la operación, pero éste no puede estar seguro que se mensaje llegó. Luego se necesitaría una cadena de infinitos mensajes para lograrlo.

41. ¿Cuál es el MTU de una red Ethernet?

El MTU de Ethernet está definido por el estándar y es de 1500 bytes + header (18 Bytes) (no es función de la red, como lo es el mínimo a transferir de 64B). Esta medida se tomó para evitar que un nodo acapare el medio más tiempo del recomendable.

42. Se dice que en 802.11 el mecanismo de *detección de portadora* es *virtual* (CSMA/CA) ¿Cómo se implementa? ¿En qué se diferencia de Ethernet 802.3?

(ver pregunta 25) En Ethernet cada nodo sensa una señal eléctrica en el medio de transmisión (cable). En Wireless no basta con sensar el medio ya que se producen problemas (terminal oculta, terminal expuesta) por lo que se utiliza un sensado “virtual” en la forma de paquetes RTS y CTS.
Cuando un nodo envían un RTS a su destino, todos los nodos cercanos lo reciben y así saben que el medio va a ser ocupado. Lo mismo sucede con todos los nodos cercanos al receptor cuando este envía un CTS.
Lo nodos cercanos al emisor/receptor calculan el tiempo de finalización de la transmisión en base a la longitud de datos a transmitir, provista por el RTS y el CTS.

43. Ejercicio de esquema de LAN con varios AP y un troncal, determinar velocidad de transferencia final y MACs involudcradas (ver práctica)

La velocidad final está determinada por el menor de los BW. Las MACs involucradas serán la de los nodos y la de los APs.

44. Describir protocolos Stop & Wait vs Sliding Windows.

Stop & Wait: consiste en enviar un único frame y esperar la respuesta ACK del receptor. Si el ACK no llega, la espera llega al *timeout* y se reenvía el mensaje. Para diferenciar las retransmisiones, se utiliza un número de secuencia de 1 bit.
Este protocolo tiene múltiples desventajas, entre ellas el desaprovechamiento del ancho de banda si los frames no ocupan todo el canal y durante el tiempo ocioso en que se espera el ACK.

Sliding Windos: consiste en mantener múltiples mensajes "en vuelo" al mismo tiempo, aprovechando el BW y controlar la recepción en orden de los mismos mediante el buffereo. Se implementa del lado del emisor y receptor una *ventana de emisión* SWS y de recepción RWS, junto a contadores que indican el último mensaje enviado, el último confirmado con ACK y, del lado del receptor, el último recibido en orden.

El emisor envía inicialmente SWS mensajes, y por cada ACK, mueve la ventana una posición, permitiendo el envío de otro mensaje. El receptor recibe los frames, posiblemente fuera de orden y los encola hasta LAF (*last aceptable frame*) número de frames., hasta que llegue el mensaje siguiente de la secuencia (LFR+1, en todo momento LAF - LAR <= RWS).
Si no llega el frame esperado, el emisor dará timeout en la espera del ACK y reenviará el frame perdido de nuevo. Para agilizar este proceso, el receptor puede enviar NACKs por cada frame fuera de orden, o ACKs repetidos con el nº de secuencia del último frame recibido en orden, de manera de permitir al emisor detectar la pérdida antes del timeout y reenviar el frame correspondiente (este mecanismo se utiliza para Fast Retransmit en TCP, ver pregunta 85).

45. ¿Cuáles son las soluciones para **controlar** la congestión de una red?

Capa 2:
En Etherner y Wireless se utiliza el algoritmo de exponential backoff al momento de transmitir.
En circuitos virtuales, reservar los recursos necesarios al establecer la conexión e ir actualizando la información durante la vida de la misma.
Capa 3: ver pregunta 63
Capa 4: ver pregunta 85

46. ¿Cuál es el throughput de un protocolo de *ventana deslizante*?

Tomando throughput como cantidad de información enviada por unidad de tiempo (no recibida) tenemos que, como una vez que un receptor lee un frame, este sale de la red mientras que se envía el correspondiente ACK, el cual tarda un t=*latencia* en llegar al emisor. Es por esto que es posible enviar hasta RTT*BW bits hasta recibir el primer ACK.

Capa 3: Red (Network)

47. ¿Cuáles son las dos grandes *estrategias* de transmisión a nivel 3? Describirlas

Datagramas: Estrategia sin conexión. Cada paquete contiene toda la información necesaria para llegar a destino (src dir, dest dir, etc). Se basa en la estrategia *best-effort*, donde los routers envían el paquete de la "mejor forma posible" utilizando las tablas de ruteo, pero donde no hay garantías de que llegue a destino, ni de que dos paquetes consecutivos sigan la misma ruta, pudiendo eventualmente llegar fuera de orden.

Las ventajas de esta transmisión son

- Un nodo puede enviar un paquete en cualquier momento, a cualquier lugar sin "aviso" previo, es decir, sin saber si el receptor siquiera está disponible.
- Es tolerante a fallas de red, pudiendo redirigir paquetes por otras rutas en caso de caída de nodos.
- No requiere de recursos previamente reservados.

Por otro lado,

- cada paquete tiene mayor overhead dada la info que contiene
- puede que el receptor no esté disponible ni activo
- es susceptible a congestión si los paquetes llegan a un router cuyas colas están llenas.

Circuitos Virtuales: orientado a conexión. Requiere que las partes, y los switches intermedios, se comuniquen y abran una conexión antes de comenzar a transmitir información.
Al inicio se intercambian paquetes indicando el comienzo de una conexión. Los switches identifican los puertos de entrada y salida en la tabla de circuitos virtuales con un ID (VCI) y comunican al nodo previo esta información una vez que el nodo destino confirma la conexión.

Al momento de transmitir los frames, en el header solo se necesita insertar el identificador de circuito virtual, de manera que se reduce el overhead por paquete.
Otra ventaja de esta estrategia es que, una vez establecida la conexión, los nodos involucrados conocen suficiente de la red como para poder reservar recursos y proveer QoS y evitar congestión.

Por el otro lado, los recursos son limitados, lo que limita la cantidad de VC's activos por switch. Al mismo tiempo, la caída de cualquier nodo del circuito implica la pérdida de la conexión y la necesidad de restablecerla.

48. Describir las políticas de servicio de comunicación QoS y Best Effort

(ver preguntas 30, 32 y 45)
QoS implica asegurar, en una conexión, entrega de paquetes en orden, sin repetidos y disponibilidad de un ancho de banda dado o un delay en la entrega de paquetes acotado en todo momento.

Best Effort implica que la red hará "todo lo posible" para entregar un paquete transmitido (ej: cambios de ruta), pero sin ofrecer ninguna garantía de éxito. Esto implica que si los paquetes se pierden, tienen errores, se corrompen, etc, la red no hace nada para solucionarlo (excepto el control de errores). También pueden entregarse paquetes fuera de orden y/o más de una vez.

49. ¿Qué control de errores se realiza en este nivel?

Checksum (ver pregunta 29) Este control se hace a nivel de **Header**, para asegurar que no se modifican campos clave, como direcciones origen y destino u otros, asumiendo que errores en los bits de mensaje son manejado en capas inferiores. Se efectúa sumando cada palabra de 16 bits del header en complemento a 1, el cual detecta errores simples que pueden no haberse detectado a nivel 2.

50. Describir e indicar ventajas y desventajas del algoritmo *Link-State*

Ventajas:
- Estabiliza rápidamente.
- Los cambios en la topología de la red se informan rápidamente: Cada router tiene una visión **global** de la red, ya que mantiene un grafo de la red y calcula el camino mínimo entre dos nodos cualesquiera.
- Cada nodo informa lo que "sabe de seguro" (nodos conectados a él), a diferencia de lo que "aprendió de otros". Luego No sufre del problema de *count-to-infinity* de DV.
- Como se almacenan todos los caminos con distancia minima: permite ofrecer balanceo de carga al distribuir los paquetes entre los múltiples caminos posibles al destino.

Desventajas:
- Requiere mayor complejidad de cómputo: calcula shortest-path con Disjktra y todos los nodos almacenan el grafo completo (OSPF).
- Uso de recursos: Cada nodo tiene que almacenar un LSP por cada otro nodo de la red.
- Cada nodo hace *flooding* (aunque controlado) en la red para avisar de la info de sus vecinos.
- Luego no escala bien en redes de muchos nodos y/o casi conexas

51. Describir e indicar ventajas y desventajas del algoritmo *Distance-Vector*

Ventajas:
- Menor complejidad computacional: utiliza Bellman-Ford
- solo guarda una matriz (RIP).
- No sobrecarga la red con paquetes al no hacer *flooding*.
- Permite que cada nodo tenga una visión general de toda la red, sin necesitar de un sistema central de control.

Desventajas:
- Convergencia lenta
- Cada nodo tiene solo una visión **local** de la red ya que solo almacena entradas del tipo <destino, distancia, next-hop>
- *Count-to-Infinity* problema (ver pregunta 55)

52. ¿Cómo escala el algoritmo Link-State?

Este algoritmo permite definir Áreas en una red. Un router de un área dada no necesita conocer todos los routers de la red, sino solamente cómo llegar a las otras áreas (simil a VLANs y STP), restringiendo el intercambio de paquetes LSP a los routers a un área dada.

53. ¿Cómo escala dentro de un sistema autónomo el protocolo OSPF?

IDEM escalabilidad de Link State

54. Describir cómo converge Distance-Vector y el problema de *conteo-a-infinito*.

- Los nodos mantienen una tabla de distancias entre si y el resto de los nodos de la red, la cual empieza con valores solo en las distancias a nodos.
- Esta info se transmite solo a los vecinos. Con lo que cada nodo pasa a actualizar las distancias a los nodos no adyacentes con la información dada por sus vecinos.
- Se actualizan los valores cuando un nodo X informa que, a través suyo, puede llegarse en menos pasos a otro nodo Y.
- Una vez que convergió el sistema (todos los nodos tienen toda la info de la red), esta información es enviada cada cierto tiempo por todos los nodos para mantener el estado. Si algún nodo detecta un fallo en la conexión con un vecino (error al transmitir o porque no se recibió el paquete de estado en el tiempo esperado), informa inmediatamente con las nuevas distancias.

El problema de count-to-infinity ocurre cuando al mismo tiempo que un nodo A informa una distancia *infinito* por la caída de un enlace con un vecino X, otro nodo B conectado a A, informa una distancia a X de 2. Como A recibe la notificación, actualiza su distancia a X a 2, lo que hace que B la actualice a 3 y así...

55. ¿Qué soluciones se proponen al problema de *conteo-a-infinito en algoritmos Link-State*? ¿y *Distance Vector*?

Link-State no sufre del problema de c-t-i porque cada nodo informa sólo lo que conoce de sus vecinos. Luego no ocurre que un nodo A informe de una ruta a C que pasa por otro nodo B.

En DV hay dos propuestas: Split Horizon y SH con Poisson Reverse. Consisten en no reenviar información acerca de un camino a aquellos nodos de los cuales se "aprendió" el camino. Más aún puede informarse distancia negativa para evitar que el otro nodo utilice esa info para actualizarse. Solo sirven para problemas de ciclos entre 2 nodos.

56. Explique por qué se dice que Link-State realiza una *inundación confiable.*

Se denomina *confiable* porque tiene las siguientes características:

- Los LSPs se envían usando transmisión confiable, mediante ACKs.
- Los LSP implementan un TTL para evitar ciclar por la red
- También tienen un número de secuencia para evitar procesar información desactualizada. Cuando un nodo recibe un LSP obsoleto, lo descarta. Si es nuevo lo procesa y reenvía.
- Los LSP se reenvían a todos **menos al nodo del cual recibió el LSP**

57. ¿Cómo se fragmentan los paquetes IP? ¿Qué relación tiene con el MTU? Describir los bits que se modifican y los que no al fragmentar.

Los paquetes IP se fragmentan a nivel de router, cuando el tamaño del paquete recibido es mayor al MTU de la red a la que debe reenviarse el datagrama (capa 2).
Para esto se marca el campo M en el header, y se define el campo Offset en 0 para el primer paquete, y luego se calcula en múltiplos de 8 Bytes. El último paquete tiene el flag M en 0.
El resto de los campos se mantienen iguales.

58. ¿Cómo implementa *calidad de servicio* IP?

Existe el campo TOS (Type of Service) que permite indicar que un paquete debe encolarse en una cola especial del router que garantiza una transmisión en tiempo acotado.

59. Describir el protocolo ARP ¿por qué se considera de nivel 3 y no de 2?

ARP permite entregar paquetes IP al nodo destino, mediante el protocolo de capa 2.
Para esto cada nodo de la LAN mantiene una tabla ARP con entradas <IP,MAC> de todos los nodos. Cuando se recibe un paquete IP con dirección destino DES, se verifica dicha tabla y se procede a colocar el paquete en la LAN con dirección MAC destino.

Si no existe la entrada, se hace broadcast de un paquete ARP *Who-Is* consultando por el nodo que tiene la IP destino. Este nodo responde con su MAC. De esta forma todos los nodos que escuchan el Who-Is también pueden actualizar su tabla ARP con la IP/MAC del emisor.

Se considera capa 3 porque administra paquetes IP.

60. ¿Qué protocolo de *Checksum* se utiliza más frecuentemente en esta capa?

Ver pregunta 49

61. Detalle el error en el siguiente razonamiento: en Packet switching se necesitan bits de control y dirección en cada paquete, lo que implica overhead, mientras que en Circuit switching solo se utiliza un Id de circuito, por lo que no existe overhead en CSw.

El *overhead* es **menor** pero existe. Para configurar un circuito virtual se necesitan múltiples mensajes entre emisor, receptor y todos los switches intermedios para definir los Ids y confirmar el establecimiento de la conexión.

62. ¿Tiene sentido el nivel 3 a nivel local?

Para pruebas de desarrollo o para montar servicios varios (bases de datos), tiene sentido poder identificar el *localhost* mediante direcciones IP (*loopback*).

63. Describir el **control** de congestión a nivel 3

Fair Queuing: el router mantiene una cola por cada conexión mantenida. Luego transmite de forma round-robin cada frame encolado, impidiendo que emisores que pueden congestionar la red, transmitan más rápido que otros. Cuando una cola está llena, se descartan los paquetes entrantes de esa conexión.

Traffic Shapping (Leaky Bucket): basado en el supuesto de que el problema de congestión se debe a las ráfagas de transmisiones no constantes. El router encola todos los mensajes y los transmite a una tasa constante, en lugar de a la tasa a la que van llegando. **Lazo abierto**.

64. Describa una red basada en conmutación de paquetes que ofrezca servicio orientado a conexión.

X.25: utiliza *sliding Windows* entre cada *hop* (router) de la ruta. De esta forma asegura entrega ordenada de paquetes entre cada segmento de la ruta completa, y por ende entre origen y destino (lo cual no es necesariamente cierto).

65. Describir los conceptos de *máscara de red* IP y *subnetting*

(ver Peterson p.302)
La técnica de *subnetting* se utiliza para, dada una única dirección IP de red (clase A, B o C), poder asignar la misma a múltiples redes físicas (sin esto, cada red física debería tener su propia IP de red).
Para esto se aprovechan los X bits de hosts de la dirección (ej: 16 bits en clase B), para separarlos en bits de *sub-red* y bits de hosts. Es entonces donde se utiliza la *máscara de subred*, para determinar, dada una dirección de la misma dirección de red, a cuál sub red física corresponde.
La máscara se define, en general, con X bits más significativos en 1's y dejando el resto en 0, de manera que, si tenemos 8 bits en 0, las direcciones posibles van desde 1 a 254 (siendo 0 local y 255 broadcast).

Cada nodo tiene entonces definida su IP y su máscara de red, mientras que los routers almacenan las tablas en formato <dirección de subred, máscara de subred, next-hop>. Cuando un host envía un paquete a una IP dada, lo primero que hace es un AND entre la IP destino y la máscara. Si el resultado pertenece a la misma subred, lo envía por capa 2, sino lo envía al router correspondiente.
Los routers hacen, para cada entrada de la tabla, un AND entre la dirección y la máscara de subred. Si el resultado coincide con la dirección de subred, se envía al nexthop.

De esta manera pueden generarse varias redes físicas o lógicas, restringiendo el espacio de broadcast, con una sola IP pública de red.

66. ¿Qué protocolos se utilizan para *multicast*?

(Ver Peterson p.330)
Broadcast: el más básico. Se transmite a todos la señal.

Link-State Multicast: funciona sobre routers que implementan OSPF. Se agrega a cada nodo del grafo de red que mantienen los routers, una etiqueta de a qué grupo pertenecen. Esto se logra haciendo que cada router informe periódicamente a qué grupo pertenece.

Luego se generan árboles de caminos-mínimos entre cada router y cada grupo multicast. Esto implica bastante costo de procesamiento y almacenamiento, por lo que se cachean.

Distance Vector Multicast: como los nodos no conocen toda la red sino el siguiente host en el camino mínimo entre dos nodos, se basa en broadcast controlado, aprovechando que se tiene entradas del tipo <destino, distancia, next-hop>. Tiene dos implementaciones:

Reverse-Path BroadCast (RPB): por cada paquete multicast que ingresa por una interfaz, genera un broadcast hacia el resto de las interfaces –no la de entrada- que tienen a la interfaz de entrada en su camino mínimo (de ahí el *reverse*).

Tiene un grave problema, que es que genera un flooding de toda la red y que no puede controlar que LANs que No están en los grupos destino, reciban los paquetes.

Reverse-Path Multicast (RPM): intenta suplir las falencias del anterior. Primero detecta, para las redes *hoja,* qué grupos desean escuchar. Los routers detectan redes hoja identificando si el router de una red es el único de la misma. Luego aumenta la entrada <destino,costo> un identificador de los grupos a los cuales la red *hoja* desea escuchar. y envía esta info a todos los routers en el camino mínimo, de manera que mensajes que no sean de esos grupos no lleguen a la red.

Protocol Independent Multicast (PIM - Sparse/Dense): protocolo que no depende del algoritmo de ruteo subyacente y que busca solucionar el problema de escalabilidad de los anteriores.
Implementa paquetes PIM para que los routers se suscriban a grupos. Estos paquetes se envían los nodos coordinadores de cada grupo (*Rendezvous Points*: RP) y estos routers RP identifican la interfaz desde donde provino el pedido y definen entonces una entrada en la tabla de ruteo multicast, indicando que todos los envíos del grupo deben reenviarse por dicha interfaz. Para conectarse por primera vez, se *tunelea* el paquete PIM en IP y se envía *unicast* al RP.
Luego los RP generan árboles de ruteo multicast para todos los destinos de su grupo correspondiente.

67. ¿Qué pasa con los paquetes que no llegan a destino después de mucho tiempo?

Al pasar por un router se decrementa el campo TTL y luego al verificar que el campo TTL está en 0, se eliminan.

68. ¿Porqué motivo OSPF tiene mayor complejidad computacional que RIP?

Porque calcula Disjktra y mantiene el grafo de toda la red en cada nodo (ver pregunta 50)

69. Explique como accede a Internet una pc vía Wireless. ¿El router usa NAT? ¿Si no tiene IP pública? ¿Si los AP tiene IP pub., hacen NAT?

- Si la PC tiene IP pública y los AP también, no se necesita NAT. Los paquetes son ruteados por el AP hacia Internet o el Gateway de salida, con la dirección pública original.
- Si la PC tiene IP privada y los AP pública, es necesario usar NAT. El AP traduce, dinámica o estáticamente, la dirección privada, por una del pool de direcciones públicas disponibles.
- Si la PC tiene IP privada y los AP también, los AP estarán conectados un router que hará NAT de la IP del AP.

70. Ejercicio de ruteo (ver de prácticas)

71. ¿Cuáles son los beneficios que introdujeron los protocolos orientados a conexión y los orientados a datagramas? ¿Cuál es actualmente más popular?

El más popular es el de Datagramas x IP, pero TCP es orientado a conexión (que NO es circuito virtual), con lo que, es debatible.

72. ¿Qué beneficios aporta la conmutación de paquetes a las comunicaciones por red?

- Aprovechamiento de recursos (routers no reservan nada, usan según necesitan)
- Interconexión de múltiples redes de diversas tecnologías y protocolos subyacentes
- Tolerancia a fallas de enlaces, permitiendo re-ruteo de paquetes.

73. ¿Cómo se comporta el tráfico de una red telefónica en comparación con el generado, por ejemplo, por conexiones http?

- El tráfico de una red telefónica es por circuito virtual, sigue siempre la misma ruta, hay recursos reservados y necesita un delay de envío de paquetes (jitter) controlado.
- Http se hace por TCP/IP, datagramas, no se asegura que cada paquete siga la misma ruta, no hay jitter controlado por la red (puede ser por las terminales) y se basa en request-response.

74. Describa el protocolo MPLS

(Ver Peterson p.341)

Protocolo que intenta mezclar la robustez de las conexiones por datagramas, con características de circuitos virtuales. Básicamente intenta:

- Proveer de capacidad de transmisión de paquetes IP a dispositivos que no tienen esta capacidad de forma natural
- Rutear paquetes IP por rutas predefinidas/precalculadas, que no necesariamente son las que tomarían en otro caso.
- Proveer soporte para ciertos tipos de VPNs

Básicamente, siguiendo la línea de los circuitos virtuales, se encapsula a los paquetes IP en un paquete MPLS con una *etiqueta* indicando el "circuito" al cual pertenece el datagrama. De esta manera, los routers pueden reenviar los paquetes por las interfaces correspondientes, solamente "mirando" el identificador de tamaño fijo, lo que hace el procesamiento más eficiente, tal como ATM x ej (de cualquier forma hoy en día no aporta performance) x la evolución de los routers.

75. Describa un algoritmo de ruteo *externo* (entre sistemas autónomos)

El algoritmo utilizado actualmente es el BGP. Se encarga de rutear entre sistemas autónomos y tiene las siguientes características:

- No asegura camino mínimo, sino que el paquete llegue a destino
- No conoce redes internas, solo SA.
- Soporta jerarquías de SAs.
- Se basa en políticas (gubernamentales, empresariales, de seguridad, etc)
- Por todo esto y por temas de escala (tablas de ruteo de todas las direcciones públicas) es muy complejo de implementar

Capa 4: Transporte (Transport)

76. ¿Cuáles son las principales diferencias entre la implementación de un servicio orientado a conexión de capa **4** con una de capa **2**?

Que en capa 4 la conexión es "virtual" en el sentido que es manejada por los nodos terminales. Los routers no conocen nada acerca de esta conexión. En capa 2 los switches conocen la existencia de la conexión.

77. Diferencias conceptuales entre protocolos de nivel 2 y 4

Punto a punto vs end-to-end
Ruteo intra-red vs entre-redes
Única dirección de envío vs multiplexación por puertos.
MTU constante, no hay fragmentación vs MTU variable / fragmentación

78. ¿Qué control de errores se realiza en este nivel? Diferencias con nivel 2

Checksum TCP: incluye **header, datos y *pseudoheader:* protocolo y direcciones origen y destino de IP**. Se implementa por Soft, por lo que el algoritmo es más sencillo y menos confiable que el CRC de capa 2.

79. ¿Para qué se usa UDP? ¿Qué información agrega el header?

Agrega info de Puertos y Checksum. Esto permite demultiplexar paquetes IP para diferentes procesos.
Un uso conocido es en consultas de DNS.

80. Describir protocolos de nivel 4: TCP, UDP

TCP (Transmission Control Protocol)

- **TCP es orientado a conexión y confiable**
- Manejo de la conexión : 3-way handshake usado para setup y 2-2 o 4 way handshake para la liberación ("problema de los dos ejércitos")
- TCP provee un servicio de flujo de bytes (stream-of-bytes): se envía un segmento cuando se llegó al total de bytes transferibles (MSS), se dio timeout en la espera de nuevos bytes o se *pusheo* el envío.
- TCP es confiable (estableciendo una suerte de "conexión lógica entre los sockets")
- Acknowledgements ACKs
- Checksums
- Números de secuencia para detectar datos perdidos o desordenados
- Datos perdidos o corruptos se RTX después de un timeout.
- Datos desordenados se podrían reordenar.
- Control de Flujo evita inundar al receptor.
- TCP implementa mecanismos de control de congestión

UDP (User Datagram Protocol):

- **No confiable, No orientado a conexión**.
- Extiende el servicio de la capa de red subyacente transformándolo en un canal de comunicación entre procesos => Agrega **demultiplexación** a IP mediante puerto origen y destino
- Aporta control de errores mediante checksum de header, datos y pseudoheader.

81. ¿Cómo se efectúa la multiplexación en TCP?

Mediante el uso de puertos. El paquete TCP contiene dos campos: Source Port y Destination Port los cuales indican a qué proceso/puerto va dirigido un mensaje para una misma dirección IP

82. ¿Cómo se establece una conexión en TCP? ¿Cómo se libera?

Establecimiento: **3-way handshake**

1. Cliente: Syn
2. Servidor: Syn + ACK
3. Cliente: ACK

Liberación: 4-way handshake

1. Cliente: Fin
2. Servidor: ACK + datos faltantes si los hay
3. Servidor: Fin
4. Cliente: ACK

83. ¿Qué es, cómo se calcula y para qué se usa el número de secuencia de un mensaje TCP?

Es el indicador del **orden** de emisión de paquetes. Se utiliza para determinar desde el lado del receptor si un paquete llegó fuera de orden, y para reordenarlo en ese caso (siempre que se encuentre dentro de la ventana de recepción).

Se define al momento de establecer la conexión. Con el último ACK se acuerda el SeqNum. De esta forma se evitan múltiple reencarnaciones de la misma conexión.
Independientemente del SeqNum elegido, el campo indica el **primer byte** del segmento: si transfiero 3KB en dos mensajes de 2KB y 1KB, entonces tenemos: SN=*k* => ACK=k+2048, SN=k+2048 => ACK= k+3072

84. ¿Para qué sirve el *puntero a urgente* en TCP?

Se utiliza para indicar un desplazamiento en bytes a partir del número de secuencia actual en el que se encuentran los datos urgentes. Esta facilidad se brinda en lugar de los mensajes de interrupción.

85. ¿Cuáles son las soluciones para **evitar** (*avoidance*) la congestión de una red?

- **DECbit (explícito, en routers)**: los routers controlan el tamaño promedio de las colas de mensajes en base a funciones del tiempo. Si el tamaño promedio es mayor a 1, entonces setean el flag DECbit en 1 para todos los paquetes enviados, el cual retorna en los ACKs enviados por el receptor. Luego el emisor chequea cuantos de sus paquetes fueron marcados con DECbit y en base al porcentaje aumenta o disminuye su ventana de congestión.
- **Random Early Detection - RED (implícito, en routers)**: este protocolo está pensado para ser usado junto a TCP. Es similar a DECBit en el hecho de que controla el tamaño de las colas y avisa cuando la congestión es inminente. El aviso en cambio, no es explícito, sino que en lugar de marcar paquetes como DECbit, descarta (drop) los paquetes que generan congestión. De esta manera el emisor disminuye la ventana de congestión *antes* de lo que lo haría por timeouts (de ahí el "early").
- **TCP Vegas (explícito, en terminales):** la fuente observa criterios que indican la presencia inminente de congestión (RTT aumenta, se aplana la curva de throughput) y decrementa la CWND linealmente.
- **Traffic Shapping (Leaky Bucket):** basado en el supuesto de que el problema de congestión se debe a las ráfagas de transmisiones no constantes. El router encola todos los mensajes y los transmite a una tasa constante, en lugar de a la tasa a la que van llegando. **Lazo abierto**.

86. Describir el mecanismo de ***Control de Congestión*** de TCP

(VER RFC)
Se basa en el concepto de **Congestion Window**, siendo una ventana de emisión de tamaño variable (a diferencia de Sliding Windows).
El proceso consta de 4 etapas:

1. **Slow Start**: se define una CW = IW = 2*SMSS y un threshold alto. Mientras CW < THRESH Se aumenta, por cada RTT, el valor de la CW en min(Datos con ACK, 1*SMSS)
2. **Congestion Avoidance**: Cuando CW > THRESH, se aumenta CWND en SMSS*SMSS/CWND por cada RTT
3. **Timeout:** al ocurrir un TO, se reduce la CW a 1xSMSS y el Threshold se setea en max(DatosEnVuelo/2, 2xSMSS) y se comienza con Slow-Start.
4. **Fast Retransmit / Fast Recovery:**
    a. si el receptor envía ACKs duplicados por cada segmento fuera de orden, el emisor asocia la recepción de 3 ACKs duplicados como indicio de pérdida de paquete, con lo que no espera hasta el TO para retransmitir.
    b. Luego de esto, pasa directamente a la etapa de Fast Recovery, seteando THRESH = max(DatosEnVuelo/2, 2xSMSS) y CWND=THRES+3 (inflado artifical de la CWND en el total de ACKs que dejaron la red y estan buffereados por el receptor).
    c. Mientras lleguen ACKs duplicados se aumenta CWND = CWND +1*SMSS
    d. En cuanto llega el primer ACK NO repetido, se setea CWND =THRESH y vuelve a Congestion Avoidance.

87. ¿Cómo se calcula el MSS (*maximum segment size*) de TCP?

Puede basase en diversos valores:

- MTU de la red del emisor
- Algoritmo MTU Path Discovery
- Receiver Window.

88. ¿Cómo se calcula, setea y ajusta el *Timeout* de retransmisión en TCP?

El cálculo de RTO se basa en el RTT, fundamental para el algoritmo de Control de Congestión, se realiza mediante ecuaciones basadas en el tiempo esperado de ACK (RTT de muestra) actual y el estimado hasta el momento. Tiene más peso el estimado que el nuevo, pero se va adaptando el valor a medida que se procesan nuevos ACKs.
Jacobson introduce el cálculo de *Desvío Estándar* del RTT con lo que el TO es el RTT + 4*DE.

Una mejora de Karn/Patridgeg consiste en No procesar ACKs repetidos y duplicar el TO luego de cada retransmisión

89. ¿Si se realiza control de flujo a nivel 2, para qué se necesita uno a nivel 4? ¿Por qué es más complejo realizarlo en este nivel?

El control de flujo a nivel 2 es punto a punto, por eso no hay relación entre emisor y receptor final, lo que impide controlar que lo que envíe el emisor, pueda ser aceptado por el receptor.

90. ¿Qué es un puerto y para qué sirve a nivel 4?

Es una dirección virtual del S.O donde procesos designados "escuchan" datos envíados a dicha dirección. Se identifican unívocamente dentro de un mismo S.O. y se asigna un puerto por proceso.
Se usa para multiplexar paquetes UDP/TCP hacia una misma dirección IP.

91. Usando TCP en un canal libre de errores y con gran ancho de banda ¿cuál es el máximo throughput obtenido?

(ver ecuación de Mathis en pregunta 102)

92. ¿Cuándo se produce una pérdida de paquetes? ¿Qué asume el soft TCP?

Cuando un router recibe más paquetes de los que puede encolar o cuando hubo un fallo en la red. TCP asume congestión.

93. ¿Qué particularidad tiene el checksum de TCP?

Que incluye el *pseudoheader* con datos de IP.

94. ¿Cómo puede efectuarse un ataque de DoS con TCP?

La forma más común es inundar al servidor con miles de pedidos SYN de conexión, de manera de bloquear las posibles sesiones de tráfico legal.

Otra forma es bloquear al emisor, inflando el CW. Se envía 1 ACK x byte y luego se pide al emisor que envíe un dato muy grande, lo que es posible porque el CW es muy grande. De esa manera bloquea el proceso por la demora en mandar mucha info.

95. ¿Cómo es posible extender la ventana del receptor en TCP?

Existe una opción en el campo Optional del header TCP.

96. Explique y fundamente qué mecanismos de TCP pueden analizarse bajo la *Teoría de Control*

Control de congestión con sus cálculos derivados: cálculo de RTT, RTO.

97. ¿Qué significa que TCP se comporta de manera *equitativa* frente a la congestión? ¿Cómo lo hace UDP?

TCP intenta disminuir la congestión, reduciendo la tasa de transferencia, mientras que UDP transfiere lo que necesita cuando lo necesita sin conocer el estado de la red, lo que puede producir más congestión.

98. Describir cómo se llega al estado *estacionario* (steady state) en TCP (Reno)

Es la punta de la curva de throughput, donde comienza un nivel bajo a medio de congestión. Se llega por aumentar constantemente la CWND en 1*SMSS*RTT en la fase Slow Start.

99. Por qué se asume un comportamiento de *dientes de sierra* para una conexión TCP?

Porque necesariamente en algún momento la red entrará en congestión (sino podría aumentar el throughput indefinidamente) y en este momento empiezan a ocurrir las pérdidas de paquetes y sus consiguientes timeouts.

100. Explicar los mecanismos de control de congestión de lazo abierto y cerrado (impl/expl)

Lazo abierto implica que el emisor No se basa en datos de la red o del receptor para determinar si ha congestión, sino que lo infiere de datos propios, como ser timeouts de transmisión. Luego decide unilateralmente cuando descartar paquetes o cuando enviarlos, sin saber nada de la red.

Lazo cerrado implica reabastecerse de datos de la red para determinar si existe congestión. Esto puede hacerse calculando RTTs entre envío y ACK, más datos del receptor o routers intermedios, como flags de congestión (DECBit).

101. Describa el funcionamiento el algoritmo RED para control de congestión

(Ver pregunta .84)

102. Dada la fórmula de Mathis *ét al. para la* performance del estado estacionario de TCP $BW = \frac{MSS \times C}{RTT \times \sqrt{p}}$ donde *p* es la probabilidad de error.

1.1.102.1 ¿Cómo se determina?

1.1.102.2 ¿qué diferencia hay entre esta ecuación con la de BW de un protocolo de nivel 2 en un enlace punto a punto sin errores?

1.1.102.3 ¿Cuál es la relación con la performance de un protocolo de nivel de enlace?

VER PAPER y diapos Claudio.

Capa 7: Aplicación (Application)

103. ¿Cómo se compone un sistema de mail?

Comienza con un User Agent componiendo un mensaje. Luego el UA se conecta, mediante SMTP sobre TCP al Mail Transfer Agent (MTA). Este consulta al DNS con la dirección destino y transfiere el mail al MTA correspondiente. Se repite el proceso hasta llegas al MTA destino.
Finalmente el receptor puede descargar el correo vía protocolos POP o IMAP.

104. Describir el protocolo SMTP-*

Un UA se conecta al puerto 25 mediante TCP y luego "habla" STMP. Se define con los comandos correspondientes el envío del mail: dirección destino, origen y cuerpo del mensaje. A cada request el Server responde con los códigos propios de HTTP. Luego se termina la sesión mediante QUIT.

105. ¿Cómo se envían datos binarios por mail?

Utilizando el protocolo MIME (base64 u otros) el cual codifica los datos en formato ASCII.

106. ¿Cuál se considera la Base de Datos distribuida más grande del planeta? ¿Por qué?

DNS. Es a grandes rasgos un gran tabla distribuida con entradas <nombre de dominio, ip> que sirve para determinar, dado un nombre de dominio de Internet, el IP del Server correspondiente. Las entradas se encuentran distribuidas por zonas y subzonas. Las zonas se distribuyen por países (.ar, .uy, etc) o de forma genéricas (.com, .edu..). Los servidores que mantienen las tablas se encuentran distribuidos alrededor del mundo según la zona que administran y se clasifican entre primarios y secundarios. Los secundarios mantienen réplicas de los primeros.
Para evitar conectarse continuamente a los DNSs externos, cada terminal cachea las entradas de forma local por un tiempo TTL determinado. Esta información se considera No-autoritativa, con lo que de ser errónea, se accede al Server primario/secundario que provee la información autoritativa/correcta.

107. ¿Por qué DNS utiliza paquetes UDP y no TCP?

DNS utiliza AMBOS tipos de paquetes para diferentes operaciones.
Para consulta de IPs dado un nombre, dado que la consulta es bien simple (clave - valor), la información transmitida muy poca y la disponibilidad de la información es alta (servers cercanos con réplicas), no es necesario establecer una conexión confiable ya que generaría mucho overhead y demoras. Basta con una operación de request-response.

Para la *transferencia de zonas*, es decir, la actualización de información de servers primarios a secundarios, SI se utiliza TCP, ya que es crítico, para el buen funcionamiento del sistema, asegurar que la información transmitida se recibió correctamente sin errores.

108. Dar una ventaja de utilizar un *servidor de nombres* contra uno de *procesos*

(Esto es fruta…)
Un servidor de procesos escucha todos los puertos de un Server y para cada mensaje entrante a un puerto X, lanza una instancia del proceso asociado a dicho puerto. De esta forma se ahorra el costo de recursos de tener múltiples procesos inactivos esperando mensajes a un puerto.

La ventaja de DNS, es que, si bien se necesita conocer el puerto al cual se dirige el pedido, no se requiere el IP del Server, con lo que simplifica el acceso y permite mudar de IP un Server sin necesidad de cambiar nada en los clientes.

Seguridad

109. ¿Cómo se aplica seguridad en las diferentes capas?

- Capa Física:
  - cables con gas a presión monitoreados, que alertan cuando son “pinchados”.
  - Técnicas de Spread Spectrum tipo DirectSequence
- Capa Enlace:
  - Encriptación WP* en wifi
- Capa Red:
  - IPSEC: encripta paquetes IP
- Capa Transporte:
  - TSL/SSL: Protege una sesión entre cliente y servidor. El caso más conocido es HTTPS (navegador y web server). Encriptación mediante clave simétrica pero definida mediante un handshake seguro con intercambio de claves pública/privada. Requiere protocolo de transporte confiable.
- Capa Aplicación:
  - SSH
  - PGP

110. Describa el algoritmo de clave *pública-privada*

La idea consiste en generar, a partir de operaciones matemáticas con números primos, dos claves complementarias, tal que un mensaje encriptado con una, solo pueda ser desencriptado con la otra. Al mismo tiempo, no debe ser computacionalmente posible descifrar una clave a partir de la otra o de un texto desencriptado.
Luego la clave *pública* está disponible para cualquiera, mientras que la privada solo debe ser conocida por el dueño. Para enviar un mensaje seguro, el emisor encripta el texto con la clave pública, y el receptor lo desencripta con la privada.
Para implementar no-repudio, puede utilizarse un mecanismode *firma-digital* donde el emisor genera un hash del texto a transmitir y lo encripta con su clave privada. Luego envía el mensaje y el hash encriptado al receptor, el cual desencripta el hash con la clave pública, lo que confirma el origen del mensaje, y verifica que el hash del texto coincida con el recibido.

111. ¿Cuáles son los conceptos principales a cumplir en términos de seguridad?

- **Autenticación**: confirmación de que un usuario/sistema es quien dice ser
- **Autorización**: control de acceso a la información
- **No-repudio**: imposibilidad de que una persona niegue haber sido parte de una comunicación/transferencia de información.
- **Confidencialidad**: ocultación de la información a terceras partes.
- **Integridad:** seguridad de que la información manejada no fue alterada.

112. ¿Cómo funciona el mecanismo de *firma digital*? ¿Qué algoritmo se utiliza?

(Ver pregunta 110)

113. Explique un mecanismo para implementar no-repudiación: emisor y receptor

Firma digital

114. ¿Existe un sistema criptográfico *perfectamente* seguro?

One-time PAD: realiza un XOR o suma x módulo del texto con una clave totalmente aleatoria de tamaño igual o mayor al texto. Esto implica que un mecanismo de descifrado sin la clave original pueda dar cualquier tipo de resultado sin precisión de su correctitud. No se implementa por razones obvias de practicidad y confidencialidad de la clave.

115. ¿En qué principio se basa un mecanismo de seguridad implementado con MD5 o SHA?

En la integridad.

116. ¿Qué puede considerarse obsoleto en el protocolo de email actual?

Transmite todo en texto plano, a menos que se use PGP.

117. ¿Porqué se dice que HTTP es un protocolo orientado a *texto?*