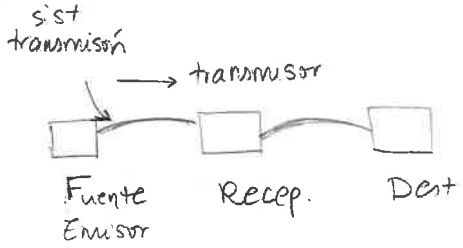
TEMA 1

red: conj equipos informáticos y software conectados por dispositivos que envían y reciben impulsos eléctricos, ondas, u otro medio para el transporte de datos para comp. información y otros servicios.

formado por
- fuente
- transmisor
- sist. transmisión
- receptor
- destino



Características red
- Autonomía – procesan infor. (main frame)
- Interconexión – mediante sistema transmisión
- Intercambio información
  - eficiente
  - transparente

Componentes
- servidor (control con funcionamiento de la red)
- est. trabajo (conj de computadoras conectadas)
- nodo red (elemento conectado a la red)
- tarjeta red (tarjetas de circuito integrado que reciben la conexión de una red).

↙ líneas transmisión ↘ nodos comunicación

Transmisión
- Coaxial (hilo cobre con una capa)
- Trenzado (conexión, 4 pares de cables)
  - UTP (bajo costo)
  - FTP (malla metálica)
  - STP (película cubre los pares)
- Fibra óptica (lo mejor pero ↑ cara)

UTP < FTP < STP < Fibra

Topología
- Física (diseño físico)
- Lógica (trayectoria señal)
  - Estrella (nodo central)
  - Árbol
  - Malla (Todos conectados con todos)
  - Híbrida (dos o + topologías)
- Bus (camino bidireccional)
- Anillo (camino unidireccional)

Clasificación redes
- LAN (broadcast)
- MAN (punto a punto)
- WAN (subred de redes)
- Internet (red de redes)
- Inalámbricas

Difusión / multipunto
- Broadcast (todo por un enlace que todos ven)
- Point-to-Point (un solo canal infor)

Gateaway: Computador especial que puede traducir infor entre sistemas con formato de datos diferentes.

TAMAÑO

TECNOLOGÍA

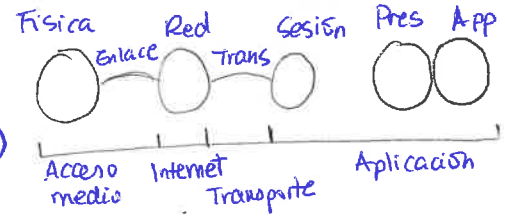
TRANSFERENCIA
- Trans. simple: solo 1 sentido
- Half Duplex: ambos sentidos, pero solo 1 en un momento dado.
- Full Duplex: ambos sentidos

Modelo ref: conjunto capas y sus funciones < de facto (No est) / de jure (estandares)

Modelo OSI

→ Capa Física (medio físico)(Tarjeta red)
→ Enlace datos (tramas, delimitación, control < errores / flujo)
→ Red (encaminamiento, control congestión, interconexión)
→ Transporte (= enlace datos, pero solo entre hosts)
→ Sesión (gestiona conexiones)
→ Presentación (representación infor)
→ Aplicación (ofrece los servicios de las otras capas)

Modelo TCP/IP



Todas las computadoras conectadas a Internet lo usan
Conjunto de protocolos que cubren los distintos niveles

Red: encaminamiento
AM: control < errores, flujo, congestión
Conexión (ext a ext)

TER Tiene SPA

Estandarización < ISO (busca estandarización)
IEE (asociación internacional sin fin de lucro)
IETF (ingeniería de Internet)

TCP/IP es una red software de facto.
Tiene una red subyacente pero varía en cada ordenador.
Entidades pantarias son aquellas del mismo nivel en emisor y receptor.
Hor. En la capa física no hay cabeceras
Protocolo = conj. reglas ; arquitectura red = capas + protocolos (p/la protocolos) < OSI X / TCP/IP V
Vert. Paso mediante la < interfaz socket ↔ (permite flujo) / SAP (información) < SDU (datos) / PDU (cabecera) entidades pantarias.

$T_T = \frac{\text{Tam (B)}}{\text{Velocidad trausm (bps)}}$ = tiempo en transferir los bits a la red.

T_p = tiempo que tarda en llegar la infor al destino = $\frac{\text{Dist (m)}}{\text{Veloc. procesamiento (m/s)}}$

$T_{procesamiento}$ = tiempo que tarda el nodo destino en leer la infor.

Tipo servicio < OT (orient. Conexion) No pérdida datos < Confirmado (correo certificado) / No confirmado
NOT (No OT) velocidad

Arquitectura red < Intranets (Ethernet): zona pública + privada redes locales
Redes acceso: xDSL, DSL, fibra óptica....
Redes troncales < Tier 1 ↔ (Gratis) / Tier 2 ↔ / Tier 3 ↔ (prestación económica)

puntos neutros son ptos donde las redes ISP intercambian tráfico
peering (ingresos generados) < acuerdos
tránsito

Direccionamiento < Nombre dominio: capa app y nombre dominio.
Dirección IP: capa red e identifica a los hosts. < públicas (unívoca) / privadas (se repiten en distintas redes)
Puertos: capa transporte y contestar peticiones

TEMA 2

Protocolo TCP/IP

Redes P2P: no hay servidores ni clientes fijos, sino nodos que son iguales.

Arpanet, ARP, RARP
→ capa red ← IPV4, IPV6
ICMP, IGMP
→ Capa transporte → TCP, UDP (extremo a extremo)
→ capa aplicación → ICMP: ping
- Telnet: terminal
- Http: paginas web.
- Smtp: correo
- Snmp: red
- bootp: arranque
- dns

TCP ← Socket → UDP

(seguro, conexión, fiable) Secuenciamiento

(veloz, No conexion, No fiable) No secuenciamiento. Util para videos y llamadas

Arq. Cliente Servidor (servidor siempre escucha)

ventajas:
- Centralización
- Seguridad
- Administración del servidor
- Escalable

Inconvenientes:
- Costo elevado
- Servidor escalón debil.

Def socket. (puerta entre app y servicios trasporte).

Socket (puntero a struct) se asocia a un puerto debe tener identificador (IP + puerto)

fiabilidad, no duplicidad } biyectividad
orden
envio mensajes urgentes
comunicacion modo conectado
conservacion limites

Tipos: Sock_Stream (1,2,3,4,6), Sock_Dgram, Sock_Raw (superusuario), Sock_seqpaquet (1,2,3,4,5)

Protocolos flexibles: cabecera pequeña añadiendo trozos (chunks)
parámetros formato TLV:
- .) 16 primeros: tipo parametro
- .) 16 siguientes: tamaño parametro (multiplo de 4B)
- .) valor

Protocolo define:
- tipo servicio ← orientado / No orientado
- tipo mensaje
- sintaxis
- semántica
- reglas (cuando los procesos envian/responden mensajes)

Tipos protocolos:
- in-band (todo por el mismo sitio)
- out-band (por un puerto se conecta por otro transmite datos)
- stateless (no retiene info)
- state full (retiene info)
- dominio publico (RFC)
- protocolos propietarios (esconder el protocolo)
- persistentes (no se cierra la conexión tras traspaso de información)
- no-persistentes (si)

Algunas caracteristicas:
- Perdida datos
- Delay · Requisitos temporales (retardo)
- Jitter · Ancho de banda, rendimiento (throughput)
- Seguridad

DNS esquema jerárquico asigna nombres significativos a grandes conjuntos de máquinas y direcciones IP. (clientes = resolvers)

Convierte nombres de dominio en direcciones IP

servicio DNS protocolo de app en la arquitectura TCP/IP tanto UDP como TCP puerto 53.

↓

crean preguntas en forma de paquetes UDP a un servidor DNS local.

Si la conexión se realiza mediante IP => No se usa DNS

Nombres dominio
- 255 caracteres
- cada dominio es un índice en BD de DNS
- Tipos
  - punto
    - absolutos ugr.
    - relativos
  - raíz
    - geográficos .es
    - organización .com

dominio genérico (.es) / subdominio (ICANN)

Servidor Root → TDp Level Domain (TLD) → Servidor Autorizado → Servidor local

Zona autoridad (parte del espacio de nombres sobre la que es responsable un servidor DNS)

Dominio (nombre que agrupa otras maquinas y dominios inferiores)

Componentes DNS
- Clientes (navegador web)
- Servidores (contestan a las consultas)
  - iterativos
  - híbrida
  - recursivos
- Zonas autoridad (almacenan datos en los servidores DNS)
  - primarios (almacenan y controlan)
  - secundarios (acceden por otro servidor)
  - maestros (transferencia zonas a los secundarios)
  - locales (solo resuelven peticiones) + cache

cliente:
interroga al servidor → interpreta respuesta → devuelve infor.

Cuando un cliente da un dominio, el DNS recibe los recursos asociados a ese nombre dominio, y debe relacionar los dominios de nombre con los registros de recursos.

- con autoridad
- sin autoridad
- no sabe la respuesta

valores
- TTL (tiempo almacenamiento)
- SOA (inicio autoridad)
- NS (nombre dominio)
- Address (dirección IP)
- MX (servidores correo en caso de fallo)

Mensajes DNS (el servidor añade una cabecera y la envía de vuelta)
- ID
- Parámetros consulta
- Nº consultas
- Nº respuestas
- Nº registros autoridad
- Nº registros adicionales

Standares Navegación Web
- URI (referencia recursos)
- HTTP (protocolo transferencia entre navegador y servidor)
- HTML (estructura y contenido documentos)
- XML (estructura documentos)

Cliente web = navegador web

determina URL, accede al DNS para, espera respuesta, se conecta al puerto
averiguar la IP (80)

envio info al servidor → post
realiza peticiones → envia un get (puede ser condicional), servidor manda, cierra conexión, se visualiza el contenido
solicitar características → head
fichero

servidor web provee de datos a navegadores.
almacena info en paginas web y protocolo Http lo entregan en formato html.
puerto 80. TCP.

If-modified-since aparece en la cabecera de http, indica que la pagina te manda un recurso solo si ha sido modificado

acepta conexión, obtiene respuesta, obtiene archivo
manda el archivo y cierra conexión

Protocolo Http (TCP/IP) Intercambios infor navegador ↔ servidor.

→ version http

Mensaje
- Solicitud (Http + parametros, cabeceras, infor adicional)
- Respuesta (Resultado + cabeceras, infor adicional)

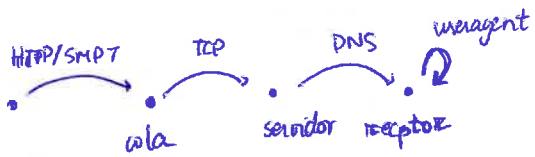
Tendencias
- Http/2 : solicitudes paralelo
- QUIC : TCP + HTTP/2 + TLS

Consistencia servidor
- → ETag
- → Last Modified
- → Cond. get → que hay en el caché

Correo electronico

Basado en SMTP y TCP. Componentes
- Cliente: una protocolo red
- Servidor
- SMTP
- Protocolo descarga: HTTP, POP3

- Usuario crea mensaje mediante su useragent
- se envia con http o smtp a la cola salientes
- El cliente abre conexión TCP con el servidor
- El cliente envia el mensaje sobre TCP.
- El servidor abre cola mensajes y los envia por DNS.
- El usuario destino invoca su useragent para leer el mensaje.



SMPT
- cliente SMPT
- servidor SMPT

→ SMTP (puerto 25) no requiere autenticación (SPAM)

→ POP3 (puerto 110) borra correo una vez descargado.

→ IMAP (puerto 143) (seguro 993)

→ HTTP (puerto 80) (seguro 443)

Primitivas seguridad
- confidencialidad
- Integridad
- Responsabilidad ← autenticación, no repudio, control accesos.
- Disponibilidad

Mecanismos seguridad
- Cifrado simétrico (misma clave cifrado que descifrado) ↑rapido ↓canal seguro donde enviar.
- Cifrado asimétrico (dos claves ≠) (Clave pública, clave privada) ↓lento.
  => se usa asimétrico para enviar las claves y luego simétrico para comunicarse.
- MAC (mensaje con un hash al final) ese hash indica la integridad.
- Firma digital: (cif. asimétrico)$^{-1}$ es un MAC con mi clave privada, y con la clave pública se puede comprobar que soy yo.
- Certificado: firma digital de un documento que asocia c.pública y c.privada.

Niveles de seguridad
- Aplicación ← SSh (protocolo crea conexión segura entre sistemas sobre redes no seguras), PGP
- Transporte ← SSL, TLS (https) (capa sesión) (En red capa enlace)
  TLS > SSL
- Red → IPSec (Seguridad IP, protocolos superiores)

Protocolos

PGP SSH	C. Aplicación	SSH (22 TCP)	FTP (21 TCP)	SMTP (25 TCP)	DNS (53 UDP/TCP)	HTTP (80 TCP)	PGP (443 S)	POP3 (110 TCP)	IMAP (143 TCP)	SNMP (161 UDP)	telnet (162 S 993 S TCP)
SSL TLS	C. Transporte	TCP, UDP, ICMP, (extremo a extremo)									
IPSEC	C. Internet	IP, IPSEC, IGMP, ICMP (datagramas) (salto a salto)									
	C Interfaz Red	Ethernet, ARP, NDP...									

Modelos Red TCP/IP, OSI (Modelos para protocolos de red)

Protocolo resolución direcciones inverso RARP

IP < IPv4, IPv6 } salto a salto. No orientado a conexión ni fiable

[MUA —SMTP— MTA — ··· —SMTP— MTA —?— MDA —HPI— MUA]

SMTP se comunican los servidores de correo

De MUA a servidor se usa HTTP, POP3, IMAP

DHCP asigna a un host una IP.

DNS permite conocer la IP.

ARP: direcciones MAC

Telnet entre maquinas remotas

SNMP intercambio información administración entre dispositivos red

FTP para pasar ficheros

SSH para acceder a maquinas remotas

DHCP asigna IP dinamica protocolo configuración

MIME permite incluir ficheros correos

Seguridad
- Firewall (SO y red) capa 7.
- Sist. detección intrusiones (IDS), red (NIDS), host (HIDS)
  - comportamiento del sistema
  - Analiza paquetes
  - busca anomalías en la act. de la máquina
- Vulnerabilidades
  - test penetración: busca vulnerabilidad del sistema
  - escanear puertos:
- Filtrado web: firewall en capa 7 (antispam)

Aplicaciones Multimedia

se usa UDP

Tipos apps multimedia
- flujo video. audio almacenado. (jitter)
- flujo video audio en vivo. (delay)
- audio video interactivo.

Características
- ↑ancho banda
- tolerante a pérdida datos
- delay acotado.
- Jitter acotado (2% delay)
- Multicast. (envío streaming a varios destinos)

Uploading: se suben los datos a una granja servidores. No hay delay.
Momentos para tareas de compresión y poder enviar el video a mas gente.

Downloading: Initial Burst (Descarga del inicio).

Aplicaciones para Interconectividad de redes locales

DHCP es el protocolo usado (genera IP aleatorias) proporciona ← DNS, IP, puerto.

1. Envío del broadcast (Busca servidor DHCP)
2. Recibe una respuesta. (También es broadcast por si hay varios servidores).
   Así conoce el puerto del servidor.
3. Solicita la IP y da gracias al resto de servidores.
4. Sigue siendo broadcast porque no se ha asignado la IP.

Otros
- UPnP: softwares P2P abren puertos
- DynDNS: detecta las modificaciones en IP y actualiza los registros DNS.

TEMA 3

Capa Transporte:
- UDP (demultiplexación (multiplexación, recibo, envío y control errores))
- TCP (UDP + notifica) Orc. Conex. (control congestión)

} Servicios

UDP (unidad es datagrama IP) (transporta TPDU)

best-effort, no fiable, No OC, no ordenado => programarlo (en capa app).
No control congestión. (RFC IETF) (Permite multiplexación, demultiplexación)

Puertos:
- 0 — 1024 (privilegios superusuario)
- 1024 — 40000
- resto

Para enviar paquetes a través del router, el router utiliza NAT para proporcionar una IP única => el puerto se lo da el router.

Wireshark:
- Nuestro ordenador solicita al DNS la dirección

Es solicitud por:
- puerto 53 (DNS)
- IP origen 172. (privada)
- query (solicitud) vs response (respuesta)

- DNS responde con la dirección
- Se conecta nuestro ordenador.

172.
192.168 ⇒ privadas

Cuando UDP recibe un paquete, mira el puerto y tiene una relación puerto-proceso donde busca el proceso correspondiente. Para comprobar los paquetes, lo que hace es sumar (en complemento a 1) todos los paquetes y aplicarle el complemento a 1, la suma debe dar 1111, en caso contrario tira el paquete. Esto es para comprobar que recibe el paquete entero.

Herramientas que usan UDP:
- DNS
- TFTP (como SFTP) (69) transferencia ficheros
- NTP (123)
- SNMP (gestión red)
- RIP (enrutamiento)

TCP (RFC) (datagrama IP)

OC, (3 pag inicio, 3/4 cerrarla), entrega ordenada, fullduplex (dos sentidos) control flujo y detección errores sofisticado. => fiable.

paquete:
- ACK (V)
- NACK (X) envío de nuevo pasado un tiempo.

No sirve para multicast fiables

También tiene control congestión y conexión
piggybacking => minimizar cantidad de paquetes enviados.

Cabecera TCP

- Puerto origen (UDP) } Cabecera UDP
- " destino (UDP) }
- Nº secuencia : identificador
- Nº "acuse" : ultimo paquete recibido por el destinatario
- HLEN: longitud cabecera
- UAPRSF: flags: 1 indica que tiene cosas especiales
  - A: piggybacking
  - P: urgente (push)
  - R: resetea la comunicación (reset)
  - S: paquete que inicia la comunicación (sincronización)
  - F: paquete que acaba la comunicación (finalizar)
- Ventana receptor: control flujo
  envío 1,2,3, espera conf del 1, envío del 4
- puntero: final del paquete urgente
- datos capa app + comprobación

El max tamaño es el min { ventana congestión, ventana receptor }

La conexión TCP se identifica por puerto IP origen y destino. (protocolo)

Herramientas que usan TCP

- FTP envío ficheros (21)
- SSH (22)
- Telnet acceso remoto (23)
- SMTP correo (25)
- DNS una TCP y UDP
- HTTP (80)
- POP3 descarga correo (110)

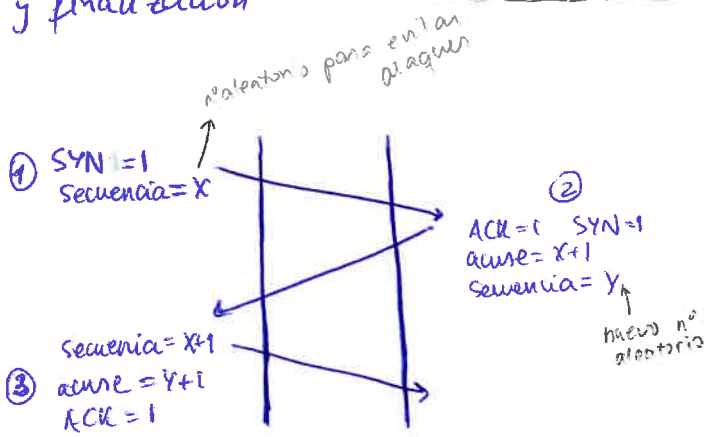
secuencia aumenta según los bytes del segmento anterior excepto cuando F=1, S=1 => aumentan en 1

Control conexión: inicio, mantenimiento y finalización

↓ three-way handshake

- servidor espera pasivamente una conexión
- el host envía una primitiva Connect, indicando el puerto de conexión. + datos ①
- el servidor recibe ese mensaje y envía otro con el proceso tcp entrante, nº secuencia ②
- Si desea conexión con el host, envía el acuse de recibo, el bit SYN y así queda abierta la conexión desde su extremo.
- el host lo recibe y envía su confirmación, momento en el que queda abierta la conexión ③

MSL: tiempo que puede existir un segmento TCP en el sistema de redes (2 min) (Max segment lifetime)



Comienza la conexión

Tras un intento fallido, lo normal es que sean distintos

Wireshark:

Los 3 primeros paquetes son de control, no hay información.

Nº acuse : siguiente byte que espera del receptor

Nº secuencia : identifica el 1er byte

Finaliza cuando envían F=1 y reciben ACK

Para terminar la conexión cualquiera de los extremos solo debe activar en un paquete el flag F. El otro responde con [acuse = secuencia+1, F=1]

Si se cierra la conexión y faltan datos, se produce four-handshake.

El nº secuencia es un campo con 32 bits. Se genera con un contador que el SO incrementa cada 4 microseg. (No sabotajes)
SYN y F incrementan el nº secuencia.

AF TCP

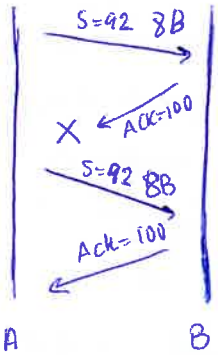
Control errores y flujo

Utiliza esquema ARQ => ACK=28 es que todo lo anterior al 28 me ha llegado bien

Campos:
- C. secuencia: offset del mensaje. Por donde voy enviando.
- C. "acuse" recibo: nº bytes esperando recepción. Por donde me confirman.
- Bit A (ACK) del campo control
- C. comprobación: checksum (igual que UDP)

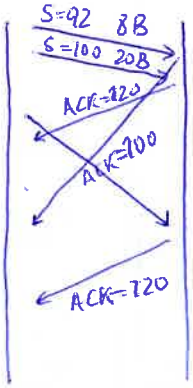
Escenarios retransmisión:
- Pérdida de ACK.
- timeout prematuro y ACK acumulativo.

• Pérdida ACK.



A manda un mensaje a B, S=92, 8B, (hasta el 99)
B responde ACK=100, el siguiente se pierde el mensaje, A no elimina los mensajes hasta recibir el ACK, para el timeout y lo vuelve a enviar.
Como B ya tenía el paquete, tira el segmento, y vuelve a enviar ACK=100.

• Timeout prematuro y ACK acum.



A manda 2 segmentos a B a la vez.
El ACK=100 se retrasa y expira el timeout, entonces se envía el segmento de nuevo como ya lo ha recibido, y ha recibido 100, 20B, envía ACK=120

- Llegada correcta, ordenada del segmento todo lo anterior confirmado => Retraso ACK 500ms (para intentar enviar menos ACK)
- Llegada ordenada del segmento, sin discontinuidad, ACK pendiente retrasado => Enviar 1 único ACK acumulativo.
- Llegada desordenada de segmento con nº sec mayor que el esperado, discontinuidad detectada => Envío ACK duplicado indicando el nº secuencia del siguiente byte
- Llegada de segmento con discontinuidad parcial o total => Confirmar ACK inmediatamente si el segmento comienza en el extremo inferior de la discontinuidad.

Estimación de los timeouts.

- Mayor que el tiempo de ida y vuelta (RTT)
- Si es pequeño => timeouts prematuros
- Si es grande => reacción lenta a pérdida de segmentos.

=> adaptable. => Con segmentos no repetidos se calcula una media, desviación, y se actualiza el timeout para ajustarlo en tiempo real.

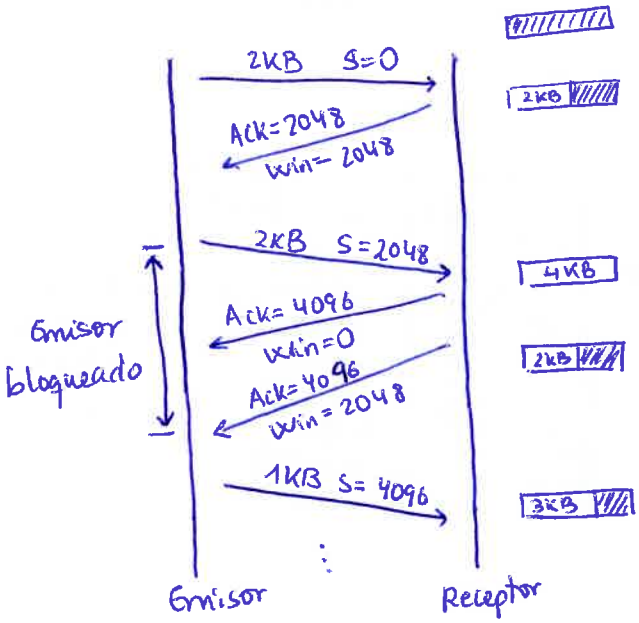
Control flujo

Otro mecanismo de control de flujo es la ventana deslizante

mecanismo que evita que emisor sature receptor.

Esquema crediticio (receptor avisa al emisor de lo que puede enviar)

El receptor, por cada ACK envía además el window, un parámetro que indica la proporción de ventana libre. util = receptor − B tránsito



Posible problema: síndrome de la ventana tonta, se debe a que el receptor avanza por el borde derecho de la ventana cada vez que tiene un nuevo espacio disponible, y el emisor utiliza cualquier ventana incremental para enviar datos. Así, el resultado puede ser un patrón de pequeños mensajes aunque ambos host tengan un gran espacio de buffer

Posible solución: ventana optimista

los segmentos urgentes (flag U) son enviados por el receptor directamente a la capa de app, igual que los que tienen el flag activado.

Push se utiliza para que TCP no espere a llenar un segmento completo.

Control congestión

Parecido al control flujo. Para estimarlo nos basamos en los "timeout", se usa para evitar que el emisor sature la red.

Tamaño ventana = MSS (Max Segment Size), y se define un umbral, define: $V_{congestion} = MSS$

- Inicio lento: $velocidad_{congestión} < umbral$.

  Envío de mensajes cada vez + rápido. Por cada ACK recibido, la velocidad congestion aumenta en 1 MSS. => crecimiento expon.

- Prevención descongestion: $velocidad_{congestión} \geq umbral$.

  se pasa a esta fase para descongestionar la red por la velocidad de inicio lento. En esta fase, por cada ventana completa (todos los ACK recibidos) aumenta el MSS pero de manera lineal.

TCP Tahoe

Si se produce timeout, estamos congestionando la red, comenzamos con $velocidad_{vent.cong} = MSS$ y el umbral = $\frac{vel\ cong}{2} = \frac{Ventana\ cong}{2}$

Eficiencia unidad es un término que indica que podemos estar todo el rato enviando segmentos si la ventana es suficientemente grande.

- Combinar control flujo con congestion se escoge la ventana más pequeña que sugieran ambos.

  B permitidos enviar = min { ventana Congestión, Ventana Receptor }
  (MMS ↑, usada control flujo ↑)

  Ventana util emisor = B permitidos enviar − B enviados.

Dependiendo del tipo de red, usaremos un TCP u otro, mención a TCP CUBIC

Una forma alternativa de estimar el RTT es enviar un segmento y que el receptor nos responde con él en el ACK. Ahí se guarda el tiempo que ha tardado, pudiendo establecer un RTT adecuado.

SACK es una confirmación selectiva.

"Sabores" TCP
- ventana escalada: opción TCP en segmentos SYN (1GB autorizado, tamaño ventana receptor 16 B
- estimación RTT: opción TCP de sello de tiempo. se sabe el tiempo que tarda el ACK.
- PAWS: cuando llega un ACK antiguo es desechado. sello de tiempo y rechazo de segmentos duplicados.
- SACK: confirmaciones selectivas. Evita duplicar envíos

TEMA 4

Protocolo IP Conmutación: determinar camino para transmitir infor ext a ext.

IPv4 interconexión redes, direccionamiento, retransmisión salto a salto, entre host/routers, No OC, no fiable, best-effort, gestiona fragmentación. ↓ adaptación a MTU's

No existe handshake, ni control < errores / flujo } = UDP ↓ pérdida datos, duplicación...

Unidad datagrama (independiente)

Para que funcione debe ∃ dirección IP asociada a nuestro ordenador.

- Máscara red: conjunto bits para delimitar el ámbito de una red. (Fija aquellos bytes que son fijos, los que varían dentro de la red no pertenecen a la máscara).

Permite enrutamiento. Divide al host de la subred.

Ej: 200.27.4.112/24 indica que los primeros 24 B, esto es, los primeros 3 nº son fijos y en esa subred lo que varía es el último número.

Dir. subred: operación AND entre IPdispositivo y máscara.

Router ≠ Subred. Switches: interconectan dispositivos (No direcciones IP

↓ Tiene IP ↓ No tiene IP Tienen dir IP los routers y hosts.

\# dispositivos = $2^{\#ceros} - 2$ ej:

200.27.4.0/24 (máscara subred / dirección de host)

- 200.27.4.0 → reservada para la subred
- 200.27.4.1
- ⋮
- 200.27.4.254
- 200.27.4.255 → reservada a broadcast (difusión)

↳ $2^8 - 2$

Direcciones
- Públicas: única en Internet
- Privadas: se pueden repetir en ≠ Intranet.

10.0.0.0/8, 172.16.0.0/11, 192.168.0.0/16

Encaminamiento: permite llevar un datagrama (paquete) de un origen a un destino. Es salto a salto.

1) routing: crear las tablas de encaminamiento en los distintos routers.

2) retransmisión: operación que hace el router. Su función es derivar los paquetes en base a la tabla encaminamiento. (Lo + típico es salto a salto)

Si queremos mandar un paquete de A a B, no se codifica todo el camino,

A → R1 → ··· → Rn → C.

plug and play: conexión sin configuración

Tabla encaminamiento [dir IP destino, máscara, siguiente Nodo]

Tabla para cada paquete, indica cual es el siguiente nodo para poder llegar a un determinado destino.

1) Cogemos la dirección destino.
2) Dirección destino & máscara. = A

¿A == dir. destino? < SI => sig. nodo
NO => seguimos buscando.

Más de una coincidencia se elige la máscara + restrictiva.
Poner la entrada por defecto.

Un SA (sistema autónomo) es un conjunto de redes y routers administrados por una autoridad, esto es, grupo de redes IP que poseen una política de rutas propias.

Niveles encaminamiento < Algoritmos IGP (protocolos dentro de un SA)
" EGP (determina si la red es accesible desde el SA)

Cabecera Datagrama IP

- Versión: determina el resto. (IPv6 se basa en chunk)
- Tamaño cabecera (20 B)
- TS (Tipo servicio)
- Identificación, I, desplazamiento.
- TTL
- Protocolo
- Checksum

ensamblar fragmentación
↙ offset ↘ identificació

Fragmentación

Fragmentación ocurre cuando un datagrama va a saltar de un segmento de red a otro en el que no cabe. (MTU)
Se vuelven a ensamblar en el destino final. (= TCP)
Existe un identificador que comparten todos los fragmentos y otro que identifica al fragmento, así:

- Si hay huecos, faltan fragmentos por llegar.
- Si No, sabemos que el último ha llegado por el flag More Fragm = 1 menos en el último que será 0.

Cabecera de IPv4 tiene al menos 20B

Conmutación (datagramas)

Determina cómo se deriva la información de un origen a un destino.

Tipos:
- Basada en circuitos (tlf) (Se reservan recursos) No hay delays pero ocupamos recursos ← Conexión, Transmisión, Desconexión
- Basada en paquetes (IP) envío en bloques y puede haber delays pero no reservamos recursos.
- Basada en paquetes con circuitos virtuales. Tecnologías intenta conmutar muy rápido. Nivel capa enlace (eficiente). Garantiza encaminamiento con mismo router, fija un camino para reducir tiempo. Recursos No dedicados.

Conmutación datagramas: no hay conexión. envío independiente. de forma que pueden llegar desordenados.

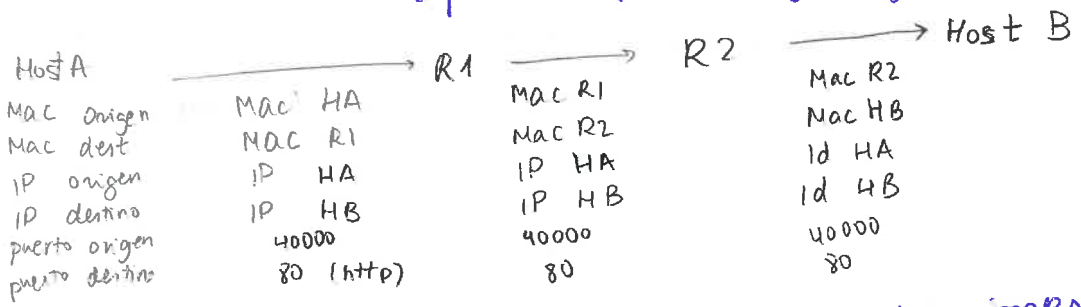
Protocolo ARP

(objetivo: obtener dir. Mac a partir IP, para ello envía un paquete 11...1)

CApp mete información al socket.
CTransporte (TDP, UDP) estuvieran directamente conectadas.
CRed ∃ capa soluciona los problemas de IP.

Dir. Mac.: identificador que identifica una tarjeta o dispositivo red unívocamente. (48 bits).
Una dir. IP es un identificador único de una conexión a un dispositivo en una red. (depende del router)

Cada paquete:
- Capa Ethernet (Dir Mac origen y destino)
- Capa IP (dir IP origen y destino)
- Capa TCP (puerto origen y destino)



El protocolo ARP asocia direcciones Mac a direcciones IP.
La cabecera es de tamaño variable.

Formato ARP.

→ Htipo
→ Ptipo } (traducción tecnología hardware y protocolo)
→ Hlen – longitud hardware
→ Plen – longitud protocolo.
→ Operación a realizar.
→ Direcciones MAC < origen / destino y IP < origen / destino.

ARP es una interacción entre capa de enlace y red.

Finalmente, el protocolo ICMP es un protocolo de gestión de red. Informa sobre direcciones mal formadas, caídas, errores, etc. se encapsula dentro de IP. (los mensaje de ICMP se encapsulan dentro de IP los paquetes IP)

Cabecera: tipo(8b) codigo(8b) comprobación (16b) Tiene checksum.

↳ 4 B + copia del que ocasionó el problema