

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

Apunte Redes

Damián Aleman

July 27, 2016

Contents

1	Teoria de la Información (Claude Shannon)	4
1.1	Información	4
1.2	Entropía	4
1.3	Modelo de un Sistema de comunicaciones	5
1.3.1	Atenuación	5
1.3.2	Distorsión de retardo	6
1.3.3	Ruido	6
2	Nivel Físico	7
2.1	Introducción y Fundamentos	7
2.2	TDM (Time Division Multiplexing)	8
2.3	FDM (Frequency Division Multiplexing)	9
2.4	WDM (Wavelength Multiplexing)	9
2.5	En telefonía	9
2.6	Taxonomía de las redes	9
2.7	Modulación	10
3	Nivel de Enlace	12
3.1	Stop And Wait	13
3.2	Sliding Window	13
3.3	Protocolos de Acceso Multiple - Medios Compartidos	13
3.3.1	CSMA-CD	14
3.4	Mecanismo de Exponential Back-Off	14
4	Nivel de Rede -Teorica 4	15
4.1	Circuitos Virtuales	15
4.2	Sin conexión: Datagramas	16
4.2.1	IP	16
4.2.2	IP Forwarding	17
5	Ruteo	17
5.1	Forwarding vs Routing	18
5.2	Protocolos de Ruteo Interno	18
5.3	Protocolos de Ruteo Externo	18
6	Nivel De Transporte	19
6.1	Protocolos End to End	19
6.2	TCP	19
6.3	Ventana deslizante	20
6.3.1	Lado Emisor	20
6.3.2	Lado Receptor	21
6.4	UDP	21
6.5	Retransmisión Adaptativa	21

6.5.1	Algoritmo original:	21
6.5.2	Algoritmo de Karn-Patridge	21
6.5.3	Algoritmo de Jacobson/Karels	21
6.6	Algoritmo de Nagle	22
6.7	Síndrome Silly Window	22
7	Performance	22
7.1	Midiendo la performance	22
7.2	Bufferbloat	22
8	Congestión	23
8.1	Síntomas de Congestión	23
8.2	Consideraciones	23
8.3	Control de Congestión vs Control de Flujo	23
8.4	Métricas	24
8.5	Causas	24
8.6	Control de Congestión	24
8.7	Performance de la red en función de la carga	24
8.8	Teoría de Control	25
8.9	Random Early Detection	25
8.10	Flow Random Early Detection	25
8.11	Traffic Shaping	25
8.11.1	Leaky Bucket	26
8.11.2	Token Bucket	26
8.11.3	Control de Congestión en TCP	26
9	Capa de Aplicación	26
9.1	CDN	27
9.2	BitTorrent	27
9.3	DNS	27
9.3.1	Decisiones de implementación	27
9.3.2	Componentes de DNS	28
9.4	Correo Electrónico	28
9.4.1	SMTP	28
9.4.2	POP3	28
9.4.3	IMAP	28
9.5	HTTP	29
10	Seguridad	29
10.1	Criptografía	29
10.2	Encriptación Simétrica	29
10.3	Encriptación Asimétrica	29
10.4	Distribución de claves	30
10.5	Firma Digital	30

10.6 Autenticación	30
------------------------------	----

1 Teoría de la Información (Claude Shannon)

Dos Teoremas Fundacionales:

1. Codificación para una fuente sin ruido
2. Codificación para un canal ruidoso: Describe la máxima eficiencia posible de un método de corrección de errores (codificación) frente a los niveles de ruido y de corrupción de los datos. Es decir, brinda un límite para la transmisión de bits. No dice nada sobre como implementar dicha codificación.

1.1 Información

Sea E un suceso que puede presentarse con probabilidad $P(E)$. Cuando E tiene lugar decimos que hemos recibido $I(E) = \log \frac{1}{P(E)}$ unidades de información.

Notar que si $P(\frac{1}{2})$, $I(E) = 1$ bit. Es decir, un bit es la cantidad de información obtenida al especificar una de dos posibles alternativas igualmente probables.

Si tenemos una fuente emitiendo una secuencia de símbolos pertenecientes a un alfabeto finito y fijo, $S = s_1, s_2, \dots, s_q$. Los símbolos emitidos sucesivamente se eligen de acuerdo con una ley fija de probabilidad. En la fuente más sencilla admitiremos que los símbolos emitidos son estadísticamente independientes. Llamaremos a la fuente de información fuente de memoria nula y puede describirse completamente mediante el alfabeto fuente S y las probabilidades con que los símbolos se presentan.

Llamaremos a una fuente de memoria nula, una fuente que emite una secuencia de símbolos pertenecientes a un alfabeto finito y fijo, $S = s_1, s_2, \dots, s_q$ con una ley fija de probabilidad y donde los símbolos emitidos son estadísticamente independientes. La fuente de memoria nula puede describirse completamente mediante el alfabeto fuente S y las probabilidades con que los símbolos se presentan $P(s_1), P(s_2), \dots, P(s_q)$

Puede calcularse la información media suministrada por una fuente de información nula en la forma siguiente:

La presencia de un símbolo si correspondiente a una cantidad de información igual a $I(s_i) = \log \frac{1}{P(s_i)}$ bits.

1.2 Entropía

Entropía $H(s)$ de la fuente de memoria nula:

La probabilidad de que aparezca un símbolo si es precisamente $P(s_i)$, de

modo que la cantidad media de información por símbolo de la fuente es:

$$\sum_S P(s_i) I(s_i) \text{ bits} = \sum_S P(s_i) \log(1/P(s_i)) \text{ bits}$$

Si tenemos una fuente que emite n mensajes s_i , la entropía es $\sum_{i=1}^n P(s_i) \log(1/P(s_i))$

Interpretaciones de la entropía: El valor medio ponderado de la cantidad de información del conjunto de mensajes posibles. Una medida de la incertidumbre promedio acerca de una variable aleatoria. La cantidad de información obtenida al observar la aparición de cada nuevo símbolo.

Propiedades de la entropía:

- La entropía es no negativa y se anula si y sólo si un estado de la variable es igual a 1 y el resto 0.
- La entropía es máxima (mayor incertidumbre del mensaje) cuando todos los valores posibles de la variable s son equiprobables.
Si hay n estados equiprobables, entonces $p_i = \frac{1}{n}$. Luego: $H(S) = -\sum p_i \log_2 p_i = -n(\frac{1}{n}) \log_2(\frac{1}{n}) = -(\log_2(1) - \log_2(n)) = \log_2(n) = H(S)_{max}$

Una fuente se puede extender mediante la extensión del alfabeto con una sucesión de los símbolos del alfabeto inicial.

Tenemos que $H(S^n) = nH(S)$

1.3 Modelo de un Sistema de comunicaciones

Canal sometido a ruido, limitado en potencia y en ancho de banda.

Perturbaciones en la transmisión. La señal recibida puede diferir de la señal transmitida:

Analógico - degradación de la calidad de la señal

Digital - Errores de bits

Causado por:

- Atenuación y distorsión de atenuación
- Distorsión de retardo
- Ruido

1.3.1 Atenuación

La intensidad de la señal disminuye con la distancia (depende del medio).

La intensidad de la señal recibida:

- Debe ser suficiente para que se detecte

- Debe ser suficientemente mayor que el ruido para que se reciba sin error
- Crece con la frecuencia

Ecualización: Amplificar más las frecuencias más altas.

1.3.2 Distorsión de retardo

Solo en medios guidos. La velocidad de propagación en el medio varía con la frecuencia. Para una señal limitada en banda, la velocidad es mayor cerca de la frecuencia central. Las componentes de frecuencia llegan al receptor en distintos instantes de tiempo, originando desplazamientos de fase entre las distintas frecuencias.

1.3.3 Ruido

Señales adicionales insertadas entre el transmisor y el receptor:

Térmico: Debido a la agitación térmica de los electrones

Intemodulación: Se produce por falta de linealidad en el canal

Diafonía: una señal de una línea interfiere en otra

Impulsivo: Impulsos irregulares o picos

Para un cierto nivel de ruido, a mayor velocidad C: menor período de un bit mayor tasa de error (se pueden corromper 2 bits en el tiempo en que antes se corrompía 1 bit)

En principio, si se aumenta el ancho de banda B y la potencia de señal S, aumenta la velocidad binaria C.

Pero: Un aumento del ancho de banda B aumenta el ruido

Un aumento de potencia de señal S aumenta las no linealidades y el ruido de intermodulación

Según Shannon, la velocidad binaria teórica máxima será para un canal será:
 $C_{\text{máx}} (\text{bps}) = B (\text{Hz}) \cdot \log_2 (1 + \text{SNR})$

Nyquist: $C(\text{bps}) = V \log_2 M = 2B \log_2 M = B \log_2 M^2$

Luego, no se podrá aumentar M tanto como se quiera: $M \leq \sqrt{(1 + \text{SNR})}$
 Teórica 2 La condición necesaria y suficiente para la existencia de un código instantáneo de longitudes l_1, l_2, \dots, l_q es que $\sum_{i=1}^q \exp(-li) \leq 1$ O cumpliendo la condición de los prefijos: No exista palabra que sea prefijo de otra palabra de longitud mayor.

Llamamos codificación al establecimiento de una correspondencia entre los símbolos de una fuente y los símbolos del alfabeto de un código. EN

la codificación intentaremos lograr una representación eficiente de la información (eliminando la redundancia).

Un código eficiente asigna las palabras mas cortas a símbolos más probables

Longitud media de un código: $L = \sum p_i l_i$ $L \log(r) \geq H(s)$ donde l_i es la longitud de la palabra codificada del mensaje y r es la cantidad de símbolos del alfabeto del código.

$\log(r)$ se interpreta como la cantidad promedio máxima de información de un símbolo del código.

Eficiencia del código $h = \frac{H(S)}{L \log(r)}$

Codificador óptimo es aquel que para codificar un mensaje X usa el menor número posible de bits. $H(X) = \sum p(x) \log_2 \left[\frac{1}{p(x)} \right]$ EL logaritmo de $\frac{1}{p(x)}$ representa el número de bits necesario para codificar el mensaje X en un codificador óptimo.

2 Nivel Físico

2.1 Introducción y Fundamentos

Hay dos Modelos fundamentales de arquitecturas de redes. Estos se basan en varias capas que son niveles de abstracción que brindan servicios a las capas superiores.

Se impuso la arquitecturas TCP/IP sobre el modelo OSI: ¹

Diagrama de sistema de comunicaciones: El mensaje se envía por un medio que sufre de ruido. Luego el receptor obtiene un mensaje que contiene variaciones.

Fundamentos de las Señales Son ondas electromagnéticas que se propagan a través de un medio: En el vacío a la velocidad de la luz (c) En otros medios a una velocidad menor, tomada como factor de c

La onda electromagnética es un campo eléctrico y magnético que se propaga por un medio a una velocidad que depende de este. La onda vibra a una frecuencia determinada, con un comportamiento periódico en el eje longitudinal de su propagación. EL periodo se denomina longitud de onda y se define como $\lambda = v/f$ (con c la velocidad y f la frecuencia de oscilación)

Problemas: La onda, en el caso de chocar con imperfecciones produce reflexiones. Además si el medio tiene perdidas se puede atenuar (generalmente se atenúan proporcionalmente a las distancias recorridas)

Longitud de onda: es la distancia espacial entre dos puntos correspondientes a la misma fase en dos ciclos consecutivos.

¹<http://spectrum.ieee.org/computing/networks/osi-the-internet-that-wasnt>

Ancho de banda: es el rango de frecuencias alrededor del punto de amplitud máxima, donde para cada lado tiene hasta que la frecuencia desciende 3 dB, o aproximadamente a la mitad de la amplitud.

Frecuencia de corte: frecuencia donde se produce una atenuación de 3dB.

Series de Fourier: Todas las funciones periódicas pueden expresarse como sumas de senos y cosenos.

$$f(t) = 1/2a_0 + \sum_{i=1}^{\infty} [a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)]$$

Una onda cuadrada se puede representar como una serie infinita de senoides armónicamente relacionados.

Existen varios medios de transmisión

- Por guía de onda:
 - Par trenzado de cobre
 - Coaxial
 - Red eléctrica
 - Fibra óptica
- Sin guía de onda (El espectro electromagnético):
 - Transmisión por radio
 - Transmisión por microondas
 - Transmisión por ondas infrarrojas
 - Transmisión por láser

Veamos la estructura del sistema telefónico: Objetivo: Transmitir la voz humana en una forma más o menos reconocible Componentes:

- Local loops (pares trenzados, señalización analógica)
- Troncales (fibra óptica o microondas, digital)
- Oficinas de conmutación

Debido a consideraciones económicas, las compañías telefónicas han desarrollado políticas elaboradas para multiplexar varias conversaciones sobre un único troncal físico.

2.2 TDM (Time Division Multiplexing)

Los usuarios toman turnos (en “round robin”) obteniendo periódicamente cada uno el ancho de banda completo por un período de tiempo acotado

2.3 FDM (Frequency Division Multiplexing)

El espectro de frecuencias es subdividido en canales de ancho de banda aco-
tado, que es usado a tiempo completo y exclusivo por cada usuario. Ejemplo
de FDM y TDM: la radio AM.

Aunque FDM se utiliza todavía sobre cables de cobre o canales de mi-
croondas, requiere circuitería analógica no trivial. En contraste TDM puede
ser manejado enteramente por electrónica digital, y se ha vuelto de más am-
plio uso en años recientes. TDM solo puede ser utilizado para datos digitales.

2.4 WDM (Wavelength Multiplexing)

En los medios ópticos se puede incrementar la capacidad transmitiendo di-
versas longitudes de onda por una única fibra (tipo especial de FDM) Se
llama WDM, multiplexación por división de longitud de onda.

2.5 En telefonía

Como el “local loop” produce señales analógicas, es necesario realizar una
conversión analógico/digital en la “end office”, donde todos los “local loops”
individuales se combinan sobre los “trunks” (troncales).

2.6 Taxonomía de las redes

1. Redes de conmutación de circuitos: FDM/TDM
2. Redes de conmutación de paquetes:

Redes de Circuitos Virtuales (VC) Brindan un servicio orientado
a conexión (ej. X.25, ATM, etc).

Redes de Datagramas (Internet) Brinda un servicio sin conexión.
Sin embargo el Nivel de Transporte brinda tanto servicios orien-
tados a conexión (TCP) como servicios sin conexión (UDP).
Lo que antes era una “conexión física” es ahora una “conexión
lógica” (o “sesión”).

La ventaja de las redes de conmutación de paquetes es que tienen una
división del tiempo, bajo demanda. Es decir se comparten los enlaces en-
colando los paquetes que compiten por el enlace cuando el mismo no esta
disponible.

El Teorema de Muestreo formulado por Nyquist (1924) dice:
Si queremos reconstruir una señal de componente frecuencial máxima f_m debemos muestrearla según $f_s > 2 * f_m$ llamada frecuencia de sampling (también de “muestreo”, o de “modulación”).

Conversión Analógica-Digital:

1. Se muestrea al doble de ancho de banda, obteniendo un tren de pulsos de amplitud variable
2. Se cuantifican las muestras aproximándolas mediante un número entero de n bits (aparece el error de cuantificación)

Canal PCM (Pulse Code Modulation) Las señales analógicas son digitalizadas por un dispositivo llamado CODEC (COder-DECoder), produciendo símbolos de 8 bits por muestra (en realidad uno es para señalización). El CODEC toma 8000 muestras por segundo ($125 \mu\text{seg}/\text{muestra}$) debido a que el teorema de Nyquist establece que esto es suficiente para capturar toda la información “relevante” de un canal telefónico de 4 KHz de ancho de banda. Luego, “Ancho de banda del cada canal de voz” = 64 Kbps. Como consecuencia, virtualmente todos los intervalos de tiempo en el sistema telefónico son múltiplos de $125 \mu\text{seg}$.

Módem convierte de digital a analógico y viceversa
Codec convierte de analógico a digital y viceversa

2.7 Modulación

Proceso de variación de cierta característica de una señal sin mensaje, llamada portadora, de acuerdo con una señal mensaje, llamada moduladora.

Tipos:

Moduladora Analógica/Portadora Analógica modular por amplitud o por frecuencia

Moduladora Digital/Portadora Analógica caso conocido la red telefónica
Técnicas:

Desplazamiento de Amplitud (ASK) los valores binarios se representan mediante dos amplitudes diferentes de la portadora

Desplazamiento de frecuencia (FSK) los valores binarios se representan mediante dos frecuencias diferentes de la portadora

Desplazamiento de Fase (PSK) los valores binarios se representan mediante dos fases diferentes de la portadora

Mixtas

Modulación Multinivel Se consigue una utilización más eficaz del ancho de banda si cada elemento de la señal transmitida representa más de un bit.

Moduladora Analógica/Portadora Digital Proceso llamado digitalización.
Dispositivo: codec

Métodos

- Modulación por impulsos codificados(MIC, o PCM)
- Modulación Delta: codifica solo las diferencias

Moduladora Digital/Portadora Digital Los datos binarios se transmiten codificando cada bit de datos en cada elemento de señal

NRZ No retorno a cero (NRZ) Consiste en utilizar una tensión negativa para representar un 0 y una positiva para representar un 1 Inconvenientes: para secuencias largas sin cambios se pierde el sincronismo (problemas de “clock recovery”)

NRZI No retorno a cero con inversión de unos (NRZI) Los datos se codifican mediante la presencia o ausencia de una transición al principio del intervalo de un 1 Soluciona el problema de muchos 1 consecutivos, pero no el de muchos 0 consecutivos.

Manchester Se codifica mediante una transición en la mitad del intervalo de duración del bit: de bajo a alto representa un 1 y de alto a bajo un 0 Baud Rate = 2 * Bit Rate

Manchester Diferencial Bifase Diferencial (Manchester Diferencial) La codificación de un 0 se representa por la presencia de una transición al principio del intervalo del bit y un 1 mediante la ausencia de transición.

Alternativa para mayor eficiencia Reemplazar secuencias de varios bits iguales (que dan lugar a niveles de tensión constante) por otra que proporcione transiciones para que emisor y receptor estén fielmente sincronizados (“preservar el clock”) El receptor debe identificar la secuencia reemplazada y sustituirla por la original. Ejemplo: 4B/5B

La Velocidad de Modulación se define como el número de cambios de señal por unidad de tiempo, y se expresa en baudios (símbolos/segundo) La Velocidad de Transmisión equivale a la velocidad de modulación multiplicado por el número de bits representados por cada símbolo, expresada en bits/segundo: $V_t = V_m N$

El precio a pagar en las modulaciones de orden superior por la mejora en la Velocidad de Transmisión es una mayor Tasa de Errores.

3 Nivel de Enlace

Enlaces Punto a Punto Tenemos un “caño” serial (no hay desordenamiento) Pero: sujeto a ruido y fallas Lo que se recibe puede no ser lo que se envió: “error de transmisión” Objetivos:

- Proveer servicio a la capa superior
- Confiabilidad. ¿Confiable o no confiable?
- Control de Errores. ¿Se produjo algún error? ¿Qué hacemos con los errores?
- Control de Flujo. Más adelante: en Nivel de Transporte.

Estrategia: Encapsulamiento o “Framing”. Encapsular los bits de Mensaje en Frames agregando información de control

Framing: ¿Cómo se separan los frames en un tren de bits?

- Largo fijo
- Largo especificado en el encabezado
- Delimitadores de frame (con bit-stuffing)

Tipo de Servicio:

- Sin conexión y sin reconocimiento
- Sin conexión y con reconocimiento
- Orientado a conexión

Para lograr una Detección y Corrección de errores es necesario tener redundancia: Definiendo: d la mínima Distancia de Hamming entre todas las codewords de un código.

e la cantidad de bits erróneos en una transmisión dada.

Necesitamos: m bits (datos) + r bits (redundancia) = n bits (codeword)

$e + 1 \leq d$ para poder detectar

$2e + 1 \leq d$ para poder corregir

Para la confiabilidad Surge la necesidad de poder efectuar retransmisiones:

Implícitas Cuando ocurre un time-out se asume que el dato se perdió

Explícitas Mensajes de control específicos para pedir repetición de envío de datos

Transmisión Confiable:

- Problema de los dos generales
- Consecuencia: No existe un algoritmo para la confiabilidad
- Enviamos un único mensaje de reconocimiento (ACK)

3.1 Stop And Wait

Cada frame debe ser reconocido por el receptor.

Problema de las reencarnaciones: Numero de Secuencia.

Existe un tiempo de bloqueo a la espera de confirmaciones.

Para aumentar la eficiencia, es decir estar bloqueado lo menos posible se creo la técnica de Sliding Window.

3.2 Sliding Window

Idea: Llenar el canal enviando el producto delay x Vtx.

Ventana de Emisión: $SWS = \frac{V_{Tx}RTT}{|Frame|}$

Enviar según: $UltimoFrameEnviado \leq UltimoFrameReconocido + SWS$

ACKs acumulativos: RWS =1

ACKs selectivos: RWS = SWS. Se informa los frames que llegaron incluso si no es el frame esperado.

Y para distinguir reencarnaciones: #frames unívocamente identificables $\geq SWS + RWS$

Delay(retardo total): $Delay = T_{prop} + T_{tx} + T_{encol} + T_{proc}$

T_{prop} = retardo de propagación. Depende de la distancia entre los hosts.

T_{tx} = Retardo de Transmisión = Tamaño Trama / Velocidad de transmisión

Significativo para enlaces de baja velocidad (o tramas muy grandes)

T_{proc} : Tiempo requerido en analizar el encabezado y decidir a dónde enviar el paquete

T_{encol} : Tiempo en que el paquete espera en un búffer hasta ser transmitido

3.3 Protocolos de Acceso Múltiple - Medios Compartidos

Vimos que podíamos “compartir” un medio de transmisión guiado o no guiada mediante:

- TDM

- FDM
- WDM
- Contención estadística: Los sistemas en los cuales varios usuarios comparten un canal común de modo tal que puede dar pie a conflictos se conocen como sistemas de contención.

El problema del acceso a un canal es hay múltiples nodos compartiendo un medio, donde la simultaneidad no es posible.

Protocolos de acceso múltiple:

Objetivo: maximizar el numero de comunicaciones exitosas, en promedio

Asegurar average fairness (igualdad de oportunidades en promedio) entre todos los nodos

Se requiere un control descentralizado.

3.3.1 CSMA-CD

Si está libre, transmite

Si esta ocupado:

1-persistente: espera a que se libere y transmite

p-persistente espera a que se libere y transmite con probabilidad p

La lógica de recepción esta establecida en el sensado para detectar colisiones

Es necesario tener un control sobre los envíos, para saber si llegaron sin colisionar Largo mínimo de trama: Se envía hasta saber que no hubo colisión.

3.4 Mecanismo de Exponential Back-Off

Elegir un slot entre 0 y $2^k - 1$, con k la cantidad de intentos.

Esperar slot veces el RTT antes de sensar para retransmitir.

Red de área local: conjunto de estacione que comparten dominio de broadcast Debido a que las LANs pueden ser de varios tipos de tecnologías, las estaciones deben compartir esquemas de direccionamiento.

Para que pueda escalar, los switchs aprenden por que interfaces deben mandarse los mensajes en función del trafico de la LAN.

Para que no hayan ciclos en una red (que generan problemas en especial cuando se hace broadcast) se ejecuta un algoritmos para eliminarlos: el Spanning Tree Protocol.

Idea: Cada switch envía paquetes (BDPU) a sus vecinos propagando información acerca de la topología de la LAN de manera periódica.

Para lograr que las LANs escalen (el broadcast no escala), existe el enfoque de LANs virtuales, quienes permiten particionar a una LAN en varias LANs diferentes e interconectarlas.

4 Nivel de Red -Teórica 4

Un switch de datos es un dispositivo con múltiples entradas y múltiples salidas. Interconecta enlaces para formar redes más grandes. El switch permite construir redes escalables.

Dos grandes paradigmas Orientado a conexión vs Sin conexión

4.1 Circuitos Virtuales

Se requiere una fase para establecer una conexión y otra de finalización de la conexión. Los paquetes o celdas que se transmiten después de establecer la conexión utilizan siempre el mismo circuito. Analogía: llamada telefónica. Cada switch mantiene una tabla VC que tiene:

1. El puerto por el cual llega el paquete.
2. El identificador del circuito virtual (VCI) de entrada
3. El puerto por el cual debe salir el paquete
4. El identificador del circuito virtual (VCI) de salida

Normalmente debe esperarse un RTT completo mientras se establece una conexión para poder enviar el primer paquete o celda. La solicitud de conexión debe llevarse para poder enviar el primer paquete o celda, pero los demás paquetes sólo tienen un identificador muy pequeño el VCI, haciendo el overhead muy pequeño. Si un switch o un enlace falla, el circuito virtual falla y una nueva conexión debe establecerse. Establecer una conexión de antemano, permite reservar recursos en los switches (espacio en buffers).
Tipos de Conexiones

Conexión Permanente (PVC) Establecimiento: Este tipo de conexión la define y la finaliza el administrador de la red: una persona solicita a la red la creación de los registros en las tablas VC. Después de creado el circuito virtual ya se pueden enviar datos. Cierre de conexión: El administrador de la red, una persona, solicita o hace las operaciones que permitan “bajar” el circuito virtual.

Conexión por Solicitud (o conmutado) (SVC) Establecimiento: Cuando el nodo A desea enviar datos al nodo B envía un mensaje de solicitud de conexión a la red, luego el switch que la recibe se lo envía al siguiente, hasta

llegar al nodo B. Este último, si acepta la conexión, devolverá el identificador de circuito que desea utilizar (4 en el ejemplo anterior) y esta “aceptación” se repite en todos los switches que se encuentran en el camino. Después de construir el circuito virtual se empieza a enviar datos. Cierre de conexión: Cuando el nodo A no desea enviar más datos al nodo B, termina el circuito virtual enviando un mensaje de finalización a la red. El switch que recibe el mensaje borra la línea de la tabla de VC correspondiente a ese circuito y envía un mensaje de finalización al siguiente switch para que repita la misma acción y así hasta alcanzar al nodo B. Si después de esto el nodo A envía un paquete o celda a la red, este puede ser descartado pues ya no existe el circuito virtual.

4.2 Sin conexión: Datagramas

El nodo puede enviar el paquete cuando quiera, no es necesario el establecimiento de una conexión. Cada paquete se envía independientemente y debe llevar toda la información necesaria para alcanzar su destino Analogía: Sistema postal. Cada switch mantiene una tabla de forwarding. Para cada nodo destino, tiene el puerto de salida que utiliza.

Ventaja: No se debe esperar un RTT para establecer una conexión. Un nodo puede enviar tan pronto como este listo. El nodo origen no tiene porque saber si la red es capaz de entregar un paquete o frame o si el nodo está listo para recibir los datos. Ya que los paquetes son tratados independientemente, es posible cambiar el camino para evitar los enlaces y los nodos que están fallando. Ya que cada paquete lleva la dirección completa del nodo destino, la información adicional de control (overhead) que lleva es mucho mayor que la utilizada en el modelo orientado a conexión.

Conmutación Source Routing Toda la información sobre la topología de la red que se necesita para conmutar los paquetes es proporcionada por el nodo origen.

4.2.1 IP

- Connectionless (datagram-based)
- Best-effort (unreliable service)
- Paquetes se pueden perder
- Enviar fuera de orden
- Entrega de copias
- No hay un cota para el tiempo de entrega

Campos del header IP

Versión: actualmente 4, comienzan los 6 pero el resto del formato del header no es el mismo en ambas versiones.

Longitud Cabecera: en palabras de 32 bits (mínimo 5, máximo 15)

Longitud total: en bytes, máximo 65535 (incluye la cabecera)

Identificación , DF, MF, Desplaz.

Fragmento: campos de fragmentación

Tiempo de vida: contador de saltos hacia atrás (se descarta cuando es cero)

Checksum: de toda la cabecera (no incluye los datos)

Dirección fuente y destino 32 bits

Cada tecnología de red tiene a nivel de enlace un MTU (Maximum Transmission Unit). IP se adapta a la tecnología de red subyacente (MTU !!) Fragmentación ocurre si un router recibe un datagrama que debe reenviar a una red donde su $MTU < |datagrama|$ Los fragmentos reciben la misma cabecera que el datagrama original salvo por los campos ‘MF’ y ‘Desplazamiento del Fragmento’. Los fragmentos de un mismo datagrama se identifican por el campo ‘Identificación’. Todos los fragmentos, menos el último, tienen a 1 el bit MF (More Fragments). La unidad básica de fragmentación es 8 bytes. Los datos se reparten en tantos fragmentos como haga falta, todos múltiplos de 8 bytes (salvo quizá el último). Toda red debe aceptar un MTU de al menos 68 bytes (60 de cabecera y 8 de datos). Recomendado 576

4.2.2 IP Forwarding

```
if NetworkNum del destino = NetworkNum de algunas de mis interfaces
then
    enviar datagrama al destino por esa interface
else
    if NetworkNum del destino esta en mi forwarding table then
        enviar datagrama al NextHop router
    else
        enviar datagrama al default router
    end if
end if
```

5 Ruteo

Los protocolos de ruteo los podemos clasificar en:

Ruteo Interno el dominio de ruteo es un sistema autónomo.

Ruteo Externo son intradominios, es decir rutean entre sistemas autónomos

5.1 Forwarding vs Routing

Forwarding debe seleccionar una puerta de salida basado en la dirección destino y las tablas de ruteo.

En cambio ruteo es el proceso mediante el cual las tablas de ruteo son construidas. Para encontrar estas tablas se debe calcular los caminos de menor costo entre los nodos. Esto se puede hacer de forma estática teniendo en cuenta solamente la topología de la red o dinámicamente teniendo en cuenta también la carga de los nodos y enlaces.

5.2 Protocolos de Ruteo Interno

	Distance-Vector	Link-State
Que informa cada nodo?	Su tabla de ruteo	Estado de sus enlaces
A quien pasa la información	Solo a sus vecinos	Inunda a toda la red
Algoritmo Utilizado	Bellman-Ford distribuido	Dijkstra
Datos Utilizados	Información de los vecinos	Estado de enlaces de cada nodo
Estructura de datos	Tabla de distancias Tabla de ruteo	Tabla de estado de enlaces Tablas de ruteo
Características	Ciclos de ruteo Gran variedad de algoritmos Cálculo distribuido	Visión consistente de enlaces Algoritmo básico único Cálculo centralizado Gran uso de CPU y memoria
Ejemplo de Internet	RIP	OSPF

En el algoritmo vector de distancia cada nodo transmite a sus vecinos lo que sabe respecto de toda la red (distancia a todos los nodos).

En el algoritmo de vectores de distancia cada nodo transmite a toda la red lo que sabe de sus vecinos. El algoritmo de estado de enlaces es estable, no genera gran tráfico y responde rápido a cambios de topología. El problema es la cantidad de información almacenada en los nodos.

En particular OSPF es escalable ya que se puede realizar de manera jerárquica con nodos troncales y nodos frontera. Por último también es más seguro (los mensajes están autenticados).

5.3 Protocolos de Ruteo Externo

Diseñado para una Internet estructurada como árbol. Se preocupa de alcanzar los nodos, no optimiza rutas. La idea es que cada sistema autónomo tiene uno más routers de borde y un portavoz BGP que publica las redes locales, otras redes alcanzables y entrega información de rutas.

6 Nivel De Transporte

Enlace de Datos versus Transporte:

- Potencialmente conecta muchas maquinas diferentes(establecimiento y termino de conexión explícitos)
- Potencialmente diferentes RTT(requiere mecanismos adaptativos para timeout)
- Potencialmente largos retardos en la red(requiere estar preparado para el arribo de paquetes muy antiguos)
- Potencialmente diferentes capacidades en destino
- Potencialmente diferentes capacidad de red

6.1 Protocolos End to End

- Garantía de entrega de mensajes
- Entrega de mensajes en el mismo orden que son enviados
- Entrega de a lo más una copia de cada mensaje
- Soporte para mensajes arbitrariamente largos mensajes
- Soporte de sincronización
- Permitir al receptor controlar el flujo de datos del transmisor
- Soportar múltiples procesos de nivel de aplicación en cada máquina

6.2 TCP

- **Orientado a conexión:** 3-way handshake para setup y 2-2 way handshake para la liberación
- **Servicio de flujo de bytes:** App escriben bytes, TCP envía segmentos, app lee bytes
- **Control de flujo:** evita que el transmisor inunde al receptor
- **Control de congestión:** evita que el transmisor sobrecargue la red
- **Full duplex**
- **Es confiable:**
 - ACKs

- Checksums
- Números de secuencia para detectar datos perdidos o desordenados
- Retransmisión de datos perdidos o corruptos después de un timeout
- Datos desordenados se podrán reordenar

Cada conexión es identificada por la 4-upla [SrcPort,SrcIPAdress, DstPort,DstAdress].

Ventana deslizante + control de flujo.

Flags: SYN, FIN, RESET, PUSH , URG, ACK.

Checksum: pseudoheader(IP) + TCP header + data.

Un segmento TCP se envía cuando:

1. Segmento full(MSS bytes)
2. No está full, pero hay un timeout
3. Pushed por la aplicación

6.3 Ventana deslizante

Se agrega a la ventana deslizante la advertised window para controlar el flujo.

Tamaño del buffer de envío: MaxSendBuffer Tamaño del buffer de recepción: MaxRcvBuffer

6.3.1 Lado Emisor

$\text{LastByteAcked} \leq \text{LastByteSent}$

$\text{LastByteSent} \leq \text{LastByteWritten}$

Se bufferean los bytes entre LstByteAcked y LastByteWritten

$\text{LastByteSent} - \text{LastByteAcked} \leq \text{AdvertisedWindow}$

$\text{EffectiveWindow} = \text{AdvertisedWindow} - (\text{LastByteSent} - \text{LastByteAcked})$

$\text{LastByteWritten} - \text{LastByteAcked} \leq \text{MaxSendBuffer}$

Bloquear Tx si $(\text{LastByteWritten} - \text{LastByteAcked}) + y \geq \text{MaxSenderBuffer}$, y bytes que se desean escribir.

Siempre enviar ACK en respuesta a la llegada de segmentos de datos

Tx persiste enviando 1 byte cuando $\text{AdvertisedWindow} = 0$

6.3.2 Lado Receptor

$LastByteRead < NextByteExpected$

$NextByteExpected \leq LastByteReceived + 1$

Se bufferean los bytes entre LastByteRead y LastByteReceived

$LastByteRcvd - LastByteRead \leq MaxRcvBuffer$

$AdvertisedWindow = MaxRcvBuffer - (LastByteRcvd - NextByteRead)$

6.4 UDP

El protocolo UDP no garantiza la confiabilidad, pero brinda la multiplexación de paquetes hacia los distintos procesos de los hosts y un checksum. Además es rápido ya que no establece una conexión.

6.5 Retransmisión Adaptativa

Dado que hay una red entre los hosts, no existe un RTT fijo.

6.5.1 Algoritmo original:

Se mide SampleRTTi por cada para segmento-ack. Calcula el promedio ponderado de RTT.

$EstimatedRTT = \alpha * EstimatedRTT + \beta * SampleRTTi$ donde $\alpha + \beta = 1$

Se fija un Timeout basado en EstimatedRTT: $Timeout = 2 * EstimatedRTT$

6.5.2 Algoritmo de Karn-Patridge

No considerar RTT cuando se retransmite Duplicar timeout luego de cada retransmisión

6.5.3 Algoritmo de Jacobson/Karels

$Diff = sampleRTT - EstRTT$

$EstRTT = EstRTT + (d * Diff)$

$Dev = Dev + d(|Diff| - Dev)$

Se considera la varianza cuando fijamos el timeout:

$Timeout = \mu * EstRTT + \phi * Dev$

Los algoritmos son tan buenos/malos como la granularidad del reloj. Es muy importante la precisa estimación del timeout para controlar la congestión. La idea es no retransmitir cuando no es necesario.

Cuando Transmitir: Envío de pocos bytes: retardar el ACK y de las actualizaciones de ventana

6.6 Algoritmo de Nagle

: If el tamaño de la ventana y los datos disponibles en el buffer \geq MSS Enviar un segmento full Else If los datos en vuelo estan sin reconocer bufferear el nuevo dato hasta que llegue el ACK Else enviar todos los datos nuevos ahora

Para que no haya problemas cuando se anuncia AdvertisedWindow = 0, TCP envía periódicamente windows probes con 1 bytes de dato

6.7 Síndrome Silly Window

Se envían datos en bloques y el receptor su app lee muy despacio , por ejemplo un 1 byte a la vez Solución: No enviar aviso de ventana para 1 Byte. Esperar hasta tener una cantidad de buffer considerable

Buffer = MSS que se anuncio en la conexión

O a la mitad de la capacidad del buffer , el valor más pequeño de los dos

7 Performance

7.1 Midiendo la performance

Hay que tener en cuenta, que a pesar de tener un bitrate físico, el medio no tiene una eficiencia total (se asume del %65). Además el bitrate tiene en cuenta a todos los datos, incluyendo los encabezados de IP, TCP.

El modelado de las transmisiones se suele hacer con una cola M/M/1. Sea R el ancho de banda del enlace (en bps), L la longitud del paquete (bits) y a la media de tasa de llegada del paquete se mide la intensidad de tráfico como λ . Cuando la intensidad es aproximadamente 0 hay un medio de retardo de cola pequeño. A medida que va creciendo la intensidad hay más retardo, en especial cuando la intensidad de tráfico (λ) es mayor a 1 llega más trabajo del que puede servirse. Este caso tiende a un retardo infinito.

7.2 Bufferbloat

Una medida para que no se pierdan los paquetes es agrandar los tamaños de los buffers. Esto puede llegar a mejorar el throughput de una sesión TCP pero a costa de un incremento de la latencia (afecta de igual manera en aplicaciones basadas en UDP). Luego el tamaño de los buffers impacta en la performance las aplicaciones. En 1985 John Nagle demuestra que el aumento del tamaño del buffer empeora el fenómeno de la congestión. El término bufferbloat se le llama a la existencia de los buffers excesivamente grande en los sistemas, en particular en los relacionados con las redes y comunicaciones.

Desde 2011 para certificar CM DOCSIS 3.0 es mandatorio soportar control de buffer. Los parámetros de control de buffer limitan la cantidad máxima de datos que pueden ser encolados por cada flujo de servicio. De esta manera se ofrece un mecanismo para balancear la relación throughput y latencia de un servicio.

8 Congestión

La congestión se define como el estado de sobrecarga sostenida de una red, donde:

La demanda de recursos (enlaces y buffers) se encuentra al límite o excede la capacidad de los mismos.

La consecuencia es perceptible en términos de QoS degradada.

Hay varias formas de encarar la solución:

- Sobredimensionamiento
- Diseño cuidadoso
- Control proactivo
- evitar (control preventivo)
- Decrementar la carga

8.1 Síntomas de Congestión

Pérdida de paquetes los buffers se saturan en routers o switches)

Retardos crecientes por las colas en los buffers

8.2 Consideraciones

Se considera que los nodos tienen una política de schedulling **FIFO** y una política de manejo de colas **Drop tail**. Ambas políticas producen una sincronización global cuando los paquetes descartados provienen desde distintas conexiones no sincronizadas entre ellas (lo cual es casi siempre).

8.3 Control de Congestión vs Control de Flujo

La congestión es un efecto global, involucra a todos los hosts y routers compartiendo una subred. Se evita que los transmisores sobrecarguen el interior de la red. En cambio el control de flujo controla el tráfico punto a punto entre un receptor y un transmisor particulares. Se evita que los transmisores sobrecarguen a receptores lentos.

8.4 Métricas

El crecimiento de alguna o varias de estas métricas indican congestión:

- Porcentaje de paquetes descartados por falta de espacio en buffer
- Longitud media de una cola (buffer)
- Cantidad de paquetes que generan timeout y son RTX
- average packet delay
- standard deviation of packet delay

8.5 Causas

- Inundo con tráfico destinado a una misma línea de salida (la cola se llena, tail drop)
- Procesadores lentos o problemas con software de ruteo
- Cuellos de botella en algunas partes del sistema (diferentes velocidades)
- El efecto de congestión tiende a realimentarse y empeorar

8.6 Control de Congestión

Es el esfuerzo hecho por los nodos de la red para prevenir o responder a sobrecargas de la red que conducen a pérdidas no controladas de paquetes. Se pueden preasignar recursos para evitar la congestión. Otra opción es la de liberar recursos y controlar la congestión sólo si ocurre (cuando ocurre).

La idea es que sea utilizada eficientemente y al mismo tiempo en forma equitativa.

Un buen indicador de la eficiencia es la *potencia* = $throughput/delay$. El control se puede implementar en los extremos de la red como en los routers dentro de ella.

8.7 Performance de la red en función de la carga

A medida que la carga de la red aumenta, el throughput se incrementa linealmente. Sin embargo, a medida que la carga alcanza la capacidad de la red los buffers en los routers comienzan a llenarse. Esto causa el incremento del tiempo de respuesta y disminuye el throughput. Una vez que los buffers de los routers comienzan a sobrecargarse ocurre la pérdida de paquetes. Bajo cargas extremas, el tiempo de respuesta tiende a infinito y el throughput tiende a cero. A esto se le llama el punto de colapso de congestión. Cuando empieza a caer rápidamente el throughput se le llama cliff.

8.8 Teoría de Control

Los algoritmos de control de congestión se pueden clasificar en lazo abierto y lazo cerrado. A su vez los de lazo cerrado se pueden clasificar de acuerdo a como realizan la realimentación: implícita (TCP) o explícita (ECN). El control de lazo abierto se ve principalmente en redes de conmutación de circuitos.

El control de lazo cerrado con feedback explícito se da por ejemplo cuando se envía un mensaje con el bit ECN (Explicit Congestion Notification) en 1. El lazo cerrado con feedback implícito se ve por ejemplo en TCP cuando se intuye que hay congestión a partir de la llegada de los acks duplicados y los timeouts.

8.9 Random Early Detection

Algoritmo de notificación implícita de inminencia de congestión. Simplemente descarta el paquete (luego en TCP habrá timeouts). Podría hacerse explícita marcando el paquete (como en ECN). El descarte es aleatorio temprano, es decir: en lugar de esperar a que se llene la cola, descarta cada paquete de entrada con alguna probabilidad de descarte cada vez que la cola exceda algún nivel de descarte.

8.10 Flow Random Early Detection

Red tiene problemas de imparcialidad ya que no es justo con las conexiones de baja velocidad.

Cuando se alcanza el umbral máximo, RED descarta los paquetes aleatoriamente. Puede darse que el paquete descartado pertenezca a un flujo que esté utilizando menos recursos de los que le correspondería en igualdad de condiciones. Cuando la longitud media de la cola está en un punto fijo dentro de los dos umbrales, todos los paquetes entrantes se descartan con la misma probabilidad. FRED soluciona estos problemas de imparcialidad manteniendo umbrales y ocupaciones del buffer para cada flujo activo. Luego es muy costoso en cuanto al alto operación en los routers. FRED debe identificar cada flujo que tenga paquetes en el buffer y actualizar la información de cada paquete.

8.11 Traffic Shaping

Cuando el tráfico es rafagoso impacta en el nivel de congestión. Traffic shaping es un método de lazo abierto que trata de guiar la congestión, forzando a los paquetes a transmitirse a una velocidad más predecible. Intenta mantener el tráfico constante regulando la tasa media de transmisión de los datos.

8.11.1 Leaky Bucket

Para entender este algoritmo se puede hacer una analogía con un balde que tiene un agujero. Cuando el balde tiene agua (datos a enviar) este transmitirá a una velocidad constante. En cambio tendrá una velocidad nula cuando el balde este vacío. El mecanismo de leaky bucket no es más que un sistema single server queing con tiempo de servicio constante y cola finita. Los paquetes llegan en cualquier instante, pero a los hosts se les permite poner solo un paquete por tick en la red. Cuando los paquetes son de diferentes tamaños, es mejor usar un número fijo de bytes por tick. En el caso de que la cola este llena, los paquetes que llegan son descartados (tail drop).

8.11.2 Token Bucket

Token bucket permite picos de tráfico cuando le llega una ráfaga grande de paquetes. Cada balde genera una cantidad de tokens por una determinada cantidad de tiempo. Para transmitir se necesita consumir un token. Si no hay, se espera. En definitiva Token bucket permite un BurstSize y Average Rate.

8.11.3 Control de Congestión en TCP

Additive increase y multiplicative decrease por RTT, de acuerdo a si hay ACK o no.

Fórmula de Mathis: $BW = \frac{MSS * C}{RTT \sqrt{p}}$

9 Capa de Aplicación

Protocolos de capa de aplicación:

- Una parte de la aplicación
- Definición de mensajes intercambiados entre las aplicaciones y las acciones tomadas en los hosts.
- Uso de servicios de comunicación proporcionados por los protocolos de capas inferiores (TCP, UDP).

Paradigma tipo cliente-servidor: El cliente inicia el contacto con el servidor y solicita un servicio. El servidor proporciona el servicio solicitado al cliente.

Direccionamiento de procesos: [IP fuente, puerto fuente, IP destino, Puerto Destino]

Para elegir un servicio de transporte adecuado para la aplicación se tiene en cuenta:

Pérdida de datos Ciertas aplicaciones pueden tolerar pérdidas(audio, video) y otras no(FTP, mail)

Temporización Teniendo en cuenta los retardos (Telefonía sobre Internet, juegos interactivos)

Ancho de Banda Algunas aplicaciones requieren un mínimo de ancho de banda (audio, video), mientras otras son más flexibles(Transferencia de archivos, mail).

9.1 CDN

Las CDNs, son redes de distribución de contenido. Funcionan como nexo entre los proveedores y los consumidores. Colocando servidores propios en los ISPs, servidores que cachean o almacenan enteramente el contenido, se aseguran de que los clientes lo accedan de manera local. Esto es importante para la distribución de video, en la cual la latencia razonable es importante para que el vídeo no se vea entrecortado.

La distribución de usuarios a servidores se realiza mediante redirecciones DNS. Cuando un usuario solicita contenido almacenado en una CDN, la dirección IP que recibe para una URL dada se resuelve teniendo en cuenta su localización geográfica y en la red. De esta manera se accederá a la copia del contenido presente en el servidor más cercano a él.

9.2 BitTorrent

BitTorrent es un protocolo para distribuir archivos en Internet en el cual los nodos que bajan el archivo contribuyen al mismo tiempo a distribuirlo, subiendo partes del archivo a otros nodos que lo están bajando.

Se define swarm al conjunto de peers bajando el archivo. Los peers que tienen el archivo completo se denominan seeds.

El tracker es un servidor encargado de mantener la información del swarm y de proveer al nodo con el conjunto inicial de peers al cual se va a conectar para comenzar a bajar el archivo.

9.3 DNS

Sistema distribuido que recibe un nombre de dominio y responde la dirección IP(en realidad es un sistema de traducción genérico).Se dejó de usar el archivos hosts que era una lista de IPs,dominio, dado que no escalaba y era propenso a errores.

9.3.1 Decisiones de implementación

Arquitectura cliente-servidor Ahorramos mantenimiento

Distribuido Evitamos fallas graves por caídas de nodos

Jerarquías de servidores de nombres Le da escalabilidad

9.3.2 Componentes de DNS

Registros DNS Mantenidos por los administradores. Entre todos forman la base de datos DNS.

Servidores DNS (primarios y secundarios) Servidores Autoritativos que tienen los datos originales.

Servidores no autoritativos Dan respuesta a consultas en redes locales.

Las consultas pueden ser iterativas o recursivas. Generalmente se cachean los Name Servers para consultas posteriores.

9.4 Correo Electrónico

La idea es tener un protocolo sencillo, los mensajes se transfieren en texto en formato ASCII. Para dar lugar a imágenes y demás se creó la extensión MIME.

Un mail se basa en un Header que contiene el from, to y subject y el cuerpo del mail. Para enviar más de un tipo de archivo se usan delimitadores como técnica de framing.

9.4.1 SMTP

Es un protocolo para mandar mails. En general nos conectamos con nuestro proveedor y luego el correo pasa por varios gateways antes de llegar al destino. Si uno no tiene el IP de la casilla de correo cacheado se realizan consultas dns para enviar el mail.

9.4.2 POP3

Protocolo orientado al cliente para descargar los correos y administrarlos localmente. La gran ventaja de este protocolo es que no hace falta estar conectado para acceder a los correos. Por otro lado la responsabilidad de backup corre por lado del usuario.

9.4.3 IMAP

Protocolo orientado al servidor. Sirve cuando se accede a la misma casilla desde distintos hosts. Permite administrar carpetas del lado de servidor y guardar estados (leídos, no leído ,etc.) Los webmails y los celulares lo utilizan.

9.5 HTTP

Es un protocolo de la capa de aplicación para enviar y recibir recursos de la web. Tiene un modelo cliente servidor.

HTTP 1.1 es persistente logrando que muchos objetos puedan ser transmitidos por una única conexión. Se suelen usar técnicas de cacheo para aprovecharse de la localidad temporal de los pedidos.

10 Seguridad

Autenticación Ninguna parte puede asumir en forma no autorizada la identidad de otra parte

Confidencialidad Los mensajes sólo deben ser leídos por las partes especificadas en la comunicación

Integridad Los datos enviados no pueden ser modificados durante su transmisión

No repudiación Ninguna de las partes puede negar haber participado en una transacción.

Autorización Los servicios brindados sólo pueden ser accedidos por usuarios autorizados

10.1 Criptografía

Cifrado por sustitución

Cifrado por Transposición

One time pad Cifrado inviolable

En los algoritmos actuales se encriptan los mensajes en bloques.

10.2 Encriptación Simétrica

Se utiliza la misma llave para encriptar que para desencriptar. Una desventaja es que si quiero que lean un mensaje cifrado muchas personas tengo que crear muchas claves.

10.3 Encriptación Asimétrica

Soluciona el problema de muchas claves. Cada uno tiene una clave privada (no comparte su secreta) y una clave pública. Un algoritmo muy conocido para generar las claves públicas y privadas es RSA.

10.4 Distribución de claves

Una forma de distribuir las claves es mediante un Key distribution center, quien tiene una clave compartida con sus clientes.

A selecciona una clave K_S y se la envía a KDC con su intención de hablar con B. El KDC desencripta este mensaje y le manda la clave de sesión a B encriptada junto con la identidad de A. Luego B entiende que A quiere iniciar una comunicación con la clave K_S .

10.5 Firma Digital

Mediante la encriptación con la clave privada se puede generar una firma digital. Generalmente se encripta luego con la clave pública del receptor para que sólo el receptor adecuado sea capaz de leer el mensaje.

Sin embargo encriptar un mensaje grande con clave publica es computacionalmente caro. La solución a esto es hashear el mensaje y encriptar el hash (de longitud corta y fija). El receptor entonces compara el hash del mensaje recibido con la función de hash (ya desencriptada).

Funciones conocidas de hash: MD5 y SHA.

HMAC hashea el mensaje mezclado con la clave de una manera standard (agregando relleno).

10.6 Autenticación

Asumiendo que A quiere establecer una comunicación con B, A le envía su identidad a B. Dado que B no puede asegurarse de que es A el que envió el mensaje le envía un challenge (número random suficientemente grande). Luego A encripta el mensaje 2 con la clave que comparte con B y se lo devuelve a B ($K_{AB}(R_B)$). Cuando B recibe este mensaje cifrado ya está seguro que proviene de A ya que es él único que conoce la clave K_{AB} . Este handshake puede ser atacado mediante un ataque por sesiones. Para solucionar esto se eligen challenges de conjuntos diferentes y con claves compartidas diferente (K_{AB} y $K_{AB'}$).