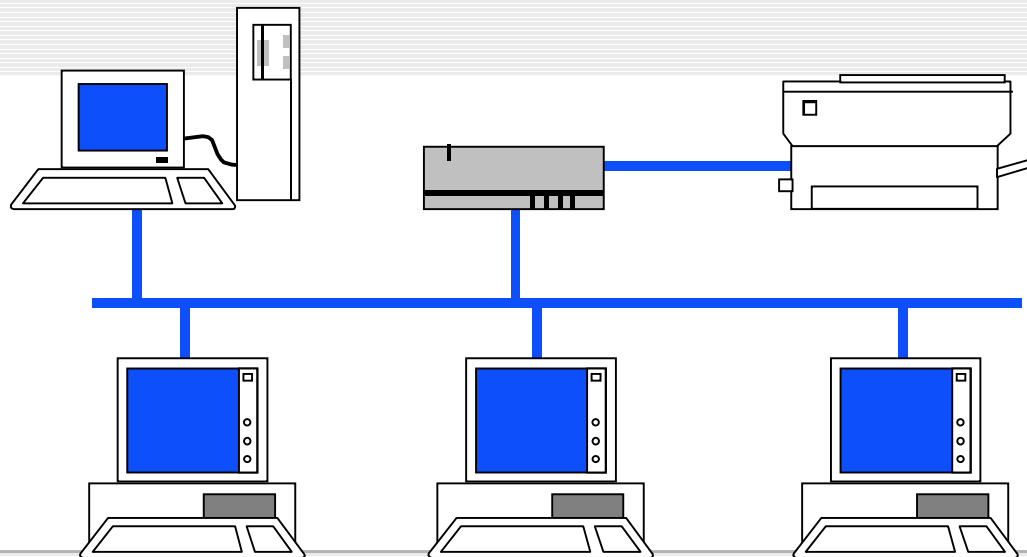
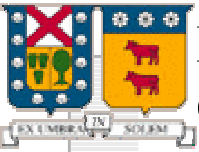


Redes de Computadores

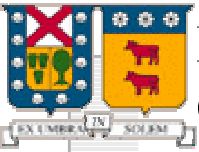
Capa de Red





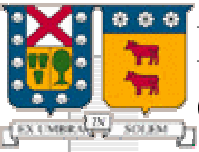
Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- **Ruteamiento basado en el Flujo**
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión

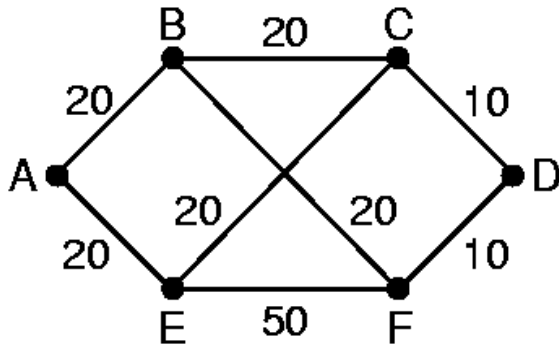


Ruteamiento basado en el flujo

- Es un método no empírico que sirve bajo ciertas condiciones que son a la vez limitadas
- Se usa en redes cuyo tráfico se puede estimar anticipadamente con una aproximación muy razonable
- Se hacen las siguientes suposiciones:
 - El tráfico es relativamente estable y predecible
 - El tráfico tiene un patrón bien definido (es simétrico)
 - El volumen total de tráfico varía muy poco entre un día y otro
- Con la ayuda de la teoría de colas se puede determinar el retardo promedio de un paquete, en cada enlace
- Se puede usar para comparar distintos algoritmos de ruteamiento (por Ej.: camino más corto)



Ruteamiento basado en el flujo

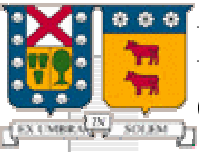


(a)

		Destination					
		A	B	C	D	E	F
Source	A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
	B	9 BA		8 BC	3 BFD	2 BFE	4 BF
	C	4 CBA	8 CB		3 CD	3 CE	2 CEF
	D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
	E	7 EA	2 EFB	3 EC	3 ECD		5 EF
	F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

Kbps

MATRIZ QUE INDICA EL TRÁFICO (Paq./Seg) Y LA MEJOR RUTA PARA CADA PAR ORIGEN-DESTINO



Ruteamiento basado en el flujo

C_i : Capacidad del enlace i en kpbs

T_i : Retardo promedio en mseg para la línea i .

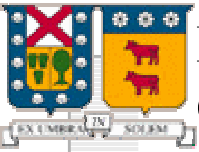
μ : Tamaño medio del paquete. Se supone $1/\mu = 800$ bits/paq.

λ_i : Tráfico total para la línea i en paq/seg

$$T_i = \frac{1}{\mu C_i - \lambda_i}$$

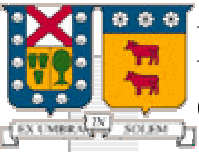
$$Weight = \frac{\lambda_i}{\sum \lambda_i}$$

i	Line	λ_i (pkts/sec)	C_i (kbps)	μC_i (pkts/sec)	T_i (msec)	Weight
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098



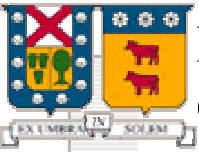
Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- **Ruteamiento Distribuido (Vector Distancia)**
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión



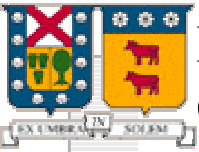
Ruteamiento Distribuido (Vector Distancia)

- ARPANET fue la primera red en utilizarlo (RIP)
- Cada Router intercambia periódicamente su tabla de rutas con sus vecinos usando paquetes broadcast o multicast
- Cada nodo mantiene una entrada por cada uno de los posibles nodos destino, la cual consta de una línea preferida y alguna estimación de la “distancia” o del retardo hacia el respectivo nodo (llamado “métrica”)
- Protocolos que son V.Distancia: RIP, RIPv2, IGRP, EIGRP

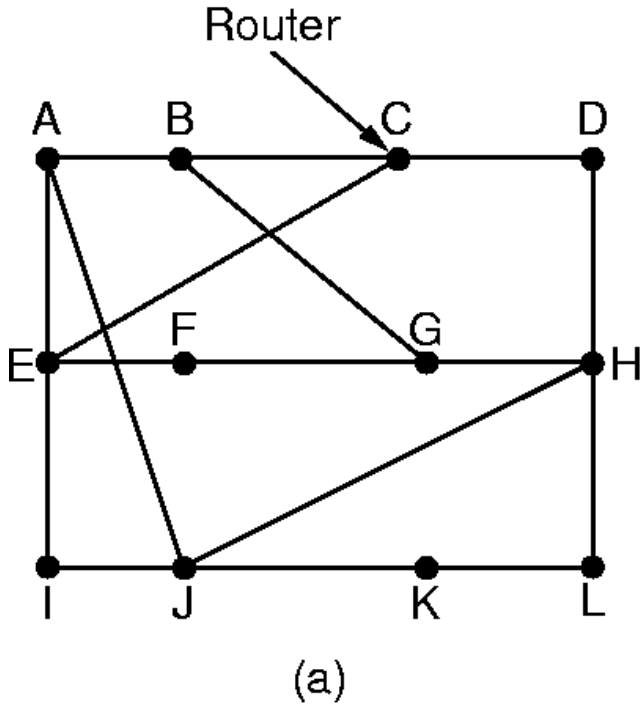


Ruteamiento Distribuido (Vector Distancia)

- Si la métrica es el retardo, cada nodo transmite un paquete especial de “eco” a sus vecinos para hacer una estimación del retardo ida-vuelta actual
- Con la información intercambiada con sus vecinos y su propia estimación, cada Router calcula los retardos para cada destino y renueva su tabla de ruteamiento
- RIP usa como métrica la cantidad de Routers (saltos)(hops) en llegar a la red destino.



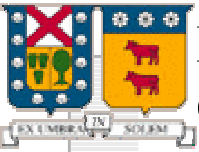
Ruteamiento Distribuido (Vector Distancia)



To	A	I	H	K	New estimated delay from J	
					↓	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	
---------------	----------------	----------------	---------------	-------------------------	--

Vectors received from J's four neighbors



Ruteamiento Distribuido (Vector Distancia)

- Problema: Conteo a infinito
 - rápida respuesta a conecciones de routers
 - respuesta lenta a desconecciones de routers

A	B	C	D	E	
•	•	•	•	•	
	∞	∞	∞	∞	Initially
1	∞	∞	∞	∞	After 1 exchange
1	2	∞	∞	∞	After 2 exchanges
1	2	3	∞	∞	After 3 exchanges
1	2	3	4	∞	After 4 exchanges

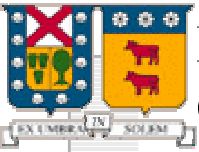
(a)

A	B	C	D	E	
•	•	•	•	•	
	1	2	3	4	Initially
3	2	3	4	4	After 1 exchange
3	4	3	4	4	After 2 exchanges
5	4	5	4	4	After 3 exchanges
5	6	5	6	6	After 4 exchanges
7	6	7	6	6	After 5 exchanges
7	8	7	8	8	After 6 exchanges
	\vdots				
∞	∞	∞	∞	∞	

(b)

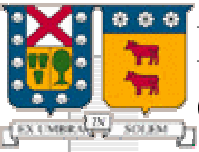
A se prende

A se apaga



Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- **Ruteamiento por estado de enlace**
- Ruteamiento Jerárquico
- Ruteamiento por Difusión

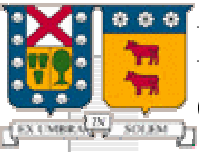


Ruteamiento por estado de enlace

Link State Routing Protocol

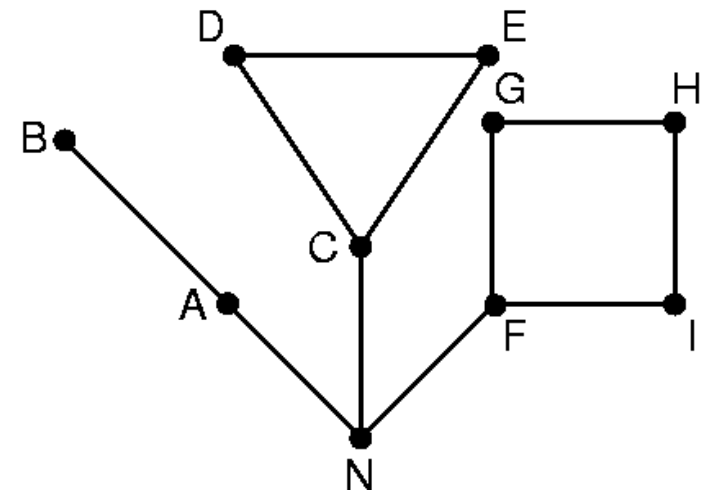
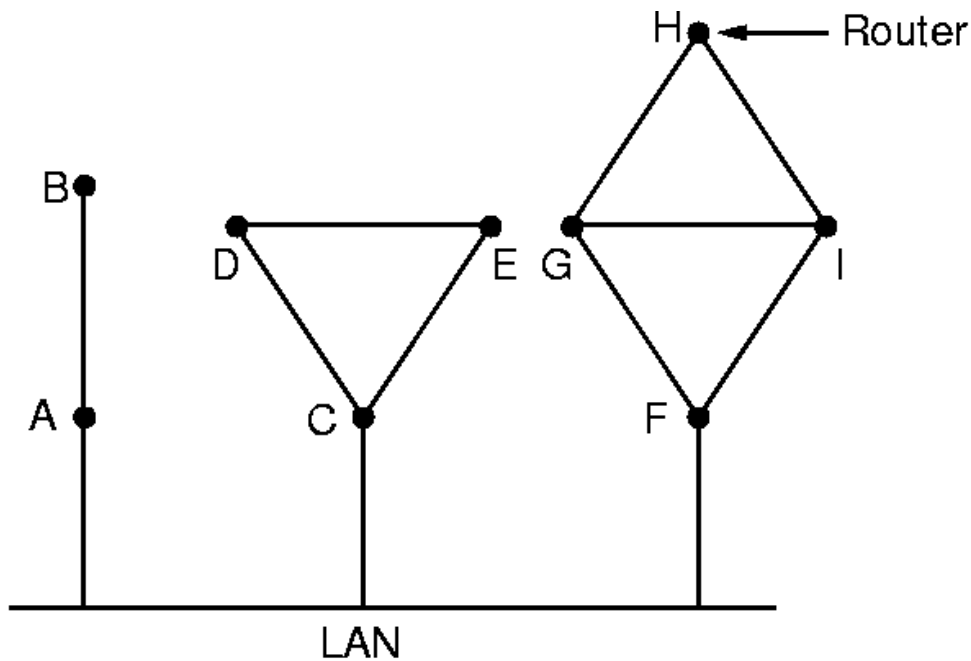
- Reemplazó al Ruteamiento por vector distancia
 - Problema del conteo a infinito (resuelto)
 - No tomaba en cuenta en BW de los enlaces

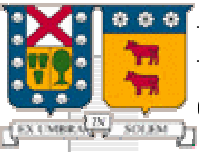
- Cada Router debe:
 - 1.- Descubrir a sus vecinos y conocer sus direcciones de red
 - 2.- Medir el retardo o costo para cada uno de sus vecinos
 - 3.- Construir un paquete que indique estas mediciones
 - 4.- Enviar este paquete a todos los demás routers
 - 5.- Calcular la trayectoria más corta a todos los demás routers. (usando Dijkstra)



Ruteamiento por estado de enlace

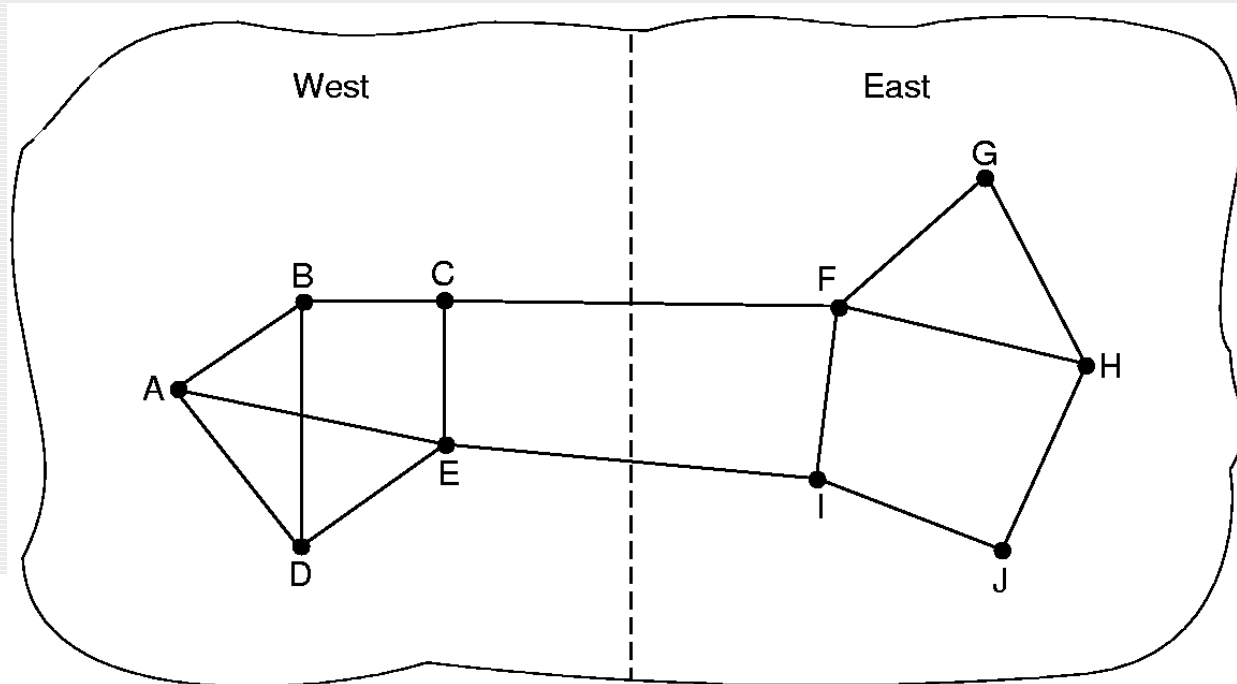
- **1.- Descubrir a sus vecinos y conocer sus direcciones de red**
 - Al encenderse envía un paquete a los otros Routers directamente conectados pidiendo información.

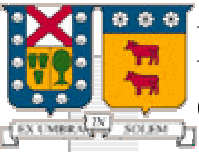




Ruteamiento por estado de enlace

- **2.- Medir el retardo o costo para cada uno de sus vecinos**
 - Cada Router envía un paquete de ECHO midiendo el tiempo de ida y vuelta.
 - Puede considerarse la “carga” del router o no....pero...
 - Si se considera, se escogerá como trayectoria más corta el enlace con menos carga.
 - Puede producir “resonancia” de enlaces...

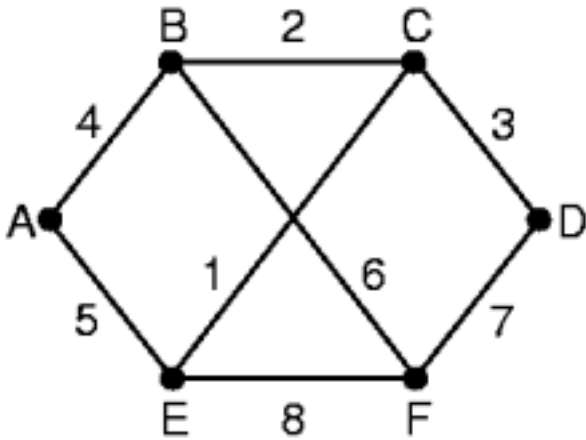




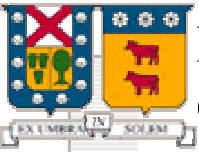
Ruteamiento por estado de enlace

3.- Construir un paquete que indique los “costos” a sus vecinos

- puede enviarlos periódicamente o cuando ocurra un evento nuevo.



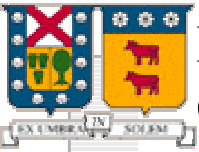
	Link	State	Packets		
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8



Ruteamiento por estado de enlace

4.- Enviar este paquete a todos los demás routers

- En principio se hace por inundación
- cada paquete lleva el número de secuencia de 32 bits que lo distingue de información antigua (a 1 pkt/s, se repite la secuencia en 137 años)
- Problema: si se cae un router y vuelve con # secuencia igual a 0, el resto descartará su información
- Problema: si ocurre un error en 1 bit justo en el # secuencia y se recibe un 65504 en vez de un 4, se descartará toda información nueva entre 5 y 65504.
- Solución: Campo de Edad. Esta se disminuye cada segundo y al llegar a cero se descarta la información.



Ruteamiento por estado de enlace

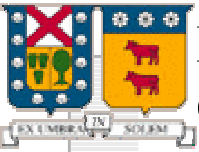
■ 5.- Calcular la trayectoria más corta a todos los demás routers. (usando Dijkstra)

- Cada router construye su grafo y ejecuta Dijkstra para encontrar la trayectoria más corta (recibe información de varias fuentes)

■ Ventaja: Sólo se intercambia información de mediciones de los Router, no toda la tabla de rutas como en Vector Distancia

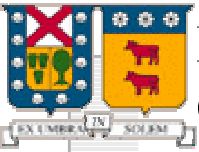
■ Este algoritmo es usado por

- OSPF (Open Shortest Path First)
- IS-IS (Intermediate System-Intermediate System)
 - CDPD
 - Novell NLSP (Ruteo de IPX)
 - Ruteo IP.



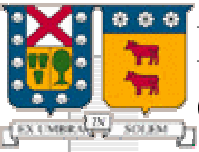
Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- **Ruteamiento Jerárquico**
- Ruteamiento por Difusión

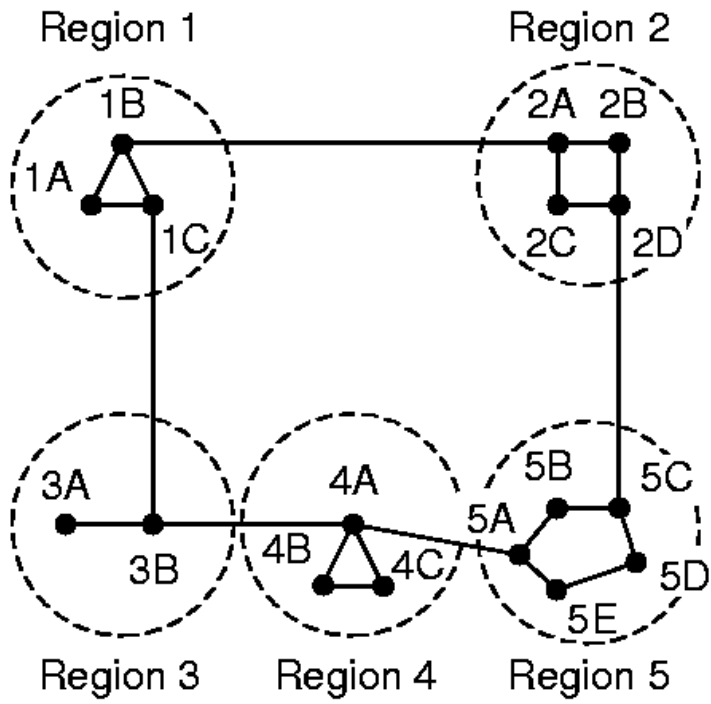


Ruteamiento Jerárquico

- Crecimiento de las Redes hace insostenible que cada Router conozca la topología y la mejor ruta a cada nodo.
- Cantidad de paquetes de información entre routers empieza a ser significativa (y el tamaño también).
- Solución:
 - Dividir la Red en regiones y sub-regiones si es necesario.
 - Conocer el Router que “maneja” una región
 - Ejemplo: Telefonía
 - Ejemplo: Internet
 - “Red de Chile” es una región distinta que “Red de Argentina”
 - Existe Super-netting.



Ruteamiento Jerárquico

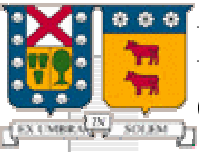


Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

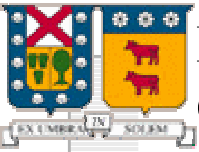
Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4



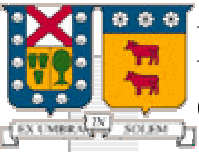
Ruteamiento Jerárquico

- Beneficios:
 - Ahorro de BW, CPU y Memoria en los Routers
- Desventajas:
 - Todo el tráfico para una región se rutea por 1 sólo camino, aunque para ciertos nodos de esa región exista una mejor ruta (o ruta alternativa)
- Cabe preguntar..Cómo debo dividir la red en forma óptima?
 - Ej: 720 Routers:
 - 24 regiones de 30 Routers (cada Router con 30+23 entradas)
 - 12 regiones de 10 zonas de 6 Routers (6+9+11 entradas)
 - Kamoun & Kleinrock (1979)
 - niveles óptimos = $\ln(N)$ (N =cantidad de Routers)
 - $e \cdot \ln(N)$ entradas por Router.



Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- **Ruteamiento por Difusión**



Ruteamiento por Difusión

Broadcast Routing

- Utilizado en aplicaciones de difusión de información
 - Actualización de Precios de la Bolsa
 - Programas de Radio en vivo
 - Informes del tiempo
- Métodos posibles:
 - broadcasting (desperdicia BW)
 - Inundación (genera muchos paquetes y consume mucho BW)
 - Ruteamiento Multidestino (utiliza Routers para replicar paquetes)
 - Spanning Tree (convertir una subred en árbol sin loops)