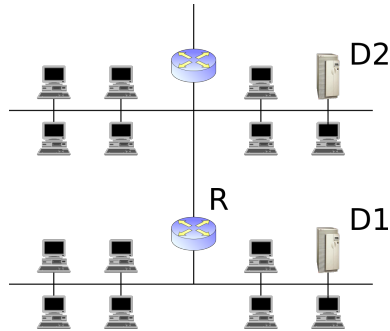


1. Tenemos parte de una red con el esquema que se adjunta.

3



Supóngase dos conjuntos de archivos de configuración para los dos servidores DHCP.

Conjunto A

D1

```
subnet 150.244.0.0 netmask 255.255.0.0 {
    range 150.244.0.1 150.244.5.80;
    option routers 150.244.3.1;
    option domain-name-server 80.58.0.33;
}
```

D2

```
subnet 150.244.0.0 netmask 255.255.0.0 {
    range 150.244.5.1 150.244.7.254;
    option routers 150.244.7.2, 150.244.7.3;
    option domain-name-server 80.58.0.33;
}
```

Conjunto B

D1

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.60 192.168.0.90;
    option routers 192.168.0.254;
    option domain-name-server 80.58.0.33;
}
```

D2

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.1 192.168.0.80;
    option routers 192.168.0.1, 192.168.0.2;
    option domain-name-server 80.58.0.33;
}
```

Indica qué problemas de red pueden producirse en los siguientes casos:

- El router R está configurado como relay de DHCP con el conjunto A de archivos de configuración para los servidores DHCP (IPs en el rango LAN). **(0.75 pts.)**
- El router R está configurado como relay de DHCP con el conjunto B de archivos de configuración para los servidores DHCP (IPs en el rango LAN). **(0.75 pts.)**
- El router no hace relay de DHCP con el conjunto A de archivos de configuración (IPs en el rango LAN). **(0.75 pts.)**
- El router no hace relay de DHCP con el conjunto A de archivos de configuración (IPs en el rango WAN). **(0.75 pts.)**

2. Imagina una red P2P propietaria, pero cuyo diseño está basado en el esquema de funcionamiento del protocolo Chord. Las funciones para el cálculo de los identificadores de nodo y contenido son las siguientes:

- $node_id() = random() \bmod 256$, donde la función $random()$ genera un número aleatorio en el rango $[0, 2^{32}]$. Si la red detecta una colisión (es decir, un nodo existente con el mismo identificador), el nodo simplemente vuelve a llamar a la función iterativamente hasta encontrar un ID disponible.
- $data_id() = b_0 \ll 3$, donde b_0 son los 4 bits menos significativos del código ASCII del primer carácter de la etiqueta del contenido a almacenar.

Responde razonadamente a las siguiente preguntas:

- ¿Cuál es el número máximo de nodos que soporta este esquema de red?. **(0,4 ptos.)**
- Si n es el número de nodos activos en la red en un momento dado, y n_{max} el número máximo de nodos soportados, ¿cuál es la probabilidad de colisión?. **(0,4 ptos.)**
- ¿Cuál es el número máximo de contenidos que puede almacenar este esquema de red?. **(0,4 ptos.)**
- Si d es el número de datos ya almacenados en la red, y d_{max} el número máximo que soporta, ¿cuál es la probabilidad de colisión en el almacenamiento de datos?. **(0,4 ptos.)**
- Imagina ahora que 4 nodos, n_i , $i=1, \dots, 4$, desean unirse a la red, para almacenar el contenido indicado más abajo, ¿cómo quedaría la distribución de nodos y contenidos en la red? La tabla $random()$ indica la salida de llamadas sucesivas a esta función. **(1 pto.)**
- Calcula, asimismo, las dos primeras entradas de la tabla de *fingers* del nodo n_3 . **(0,4 ptos.)**

Detalla al máximo la resolución del ejercicio y realiza una representación gráfica del resultado. Las cadenas del contenido son sensibles a mayúsculas y minúsculas.

$random()$	Contenido
1302	Hetzner
123	Queen
2014910	Guns 'n Roses
190	video3.mp4
56	
...	

Solución:

1. Puesto que el identificador se reduce $\bmod 256$ (aunque la función $random()$ trabaje en el rango $[0, 2^{32}]$, solo hay **256** posibles valores diferentes para el mismo.
2. n/m_{max}
3. El operador \ll corresponde al desplazamiento binario a la izquierda, que introduce ceros por la derecha del valor considerado (lo que equivale a multiplicarlo por 2). Puesto que se nos dice que se consideran los 4 bits menos significativos existen, en realidad, únicamente $2^4 = 16$ posibles valores (el operador \ll no introduce nuevos valores, solo modifica los existentes).
4. d/d_{max}
5. Veamos el caso del primero contenido a almacenar, 'Hetzner'. Para ello:

$$a) \ data_id('Hetzner') = 0x48 = 0100 \underbrace{1000}_{b_0} \ll 3 = 0x40$$

$$b) \ node_id('Hetzner') = 1302 \bmod 256 = 22$$

El resto de identificadores de nodos se calculan de la misma forma, y se ordenan de la manera habitual. Si utilizamos la notación $n_i[data_id, node_id]$ para identificar cada nodo y el contenido que almacena, quedaría:

$$n_1[0x40, 22] > n_4[0x20, 56] > n_2[0x08, 123] > n_3[0x38, 190] > n_1$$

6. La tabla de *fingers* del nodo n_3 queda:

Figura 1: Tabla ASCII

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
\$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
;	59	0073	0x3b	[91	0133	0x5b	{	123	0173	0x7b
<	60	0074	0x3c	\	92	0134	0x5c		124	0174	0x7c
=	61	0075	0x3d]	93	0135	0x5d	}	125	0175	0x7d
>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
?	63	0077	0x3f	_	95	0137	0x5f				

finger n_3

finger[0] $\text{successor}(190 + 1 \bmod 256) = n_1$

finger[1] $\text{successor}(190 + 2 \bmod 256) = n_1$