

SIECI KOMPUTEROWE – LISTA 1

Mateusz Zajac

1. Dla każdego z podanych poniżej adresów IP w notacji CIDR określ, czy jest to adres sieci, adres rozgłoszeniowy czy też adres komputera. W każdym przypadku wyznacz odpowiadający mu adres sieci, rozgłoszeniowy i jakiś adres IP innego komputera w tej samej sieci.

- 10.1.2.3/8
- 156.17.0.0/16
- 99.99.99.99/27
- 156.17.64.4/30
- 123.123.123.123/32

Jeśli chcemy wyznaczyć adres sieci, do której należy określony adres IP, musimy zapisać ten adres w postaci bitowej i porównać go z maską podsieci. Zrobię to na przykładzie pierwszego adresu:

- 10.1.2.3/8

	W postaci dziesiętnej	W postaci bitowej
Adres IP	10.1.2.3	00001010.00000001.00000010.00000011
Maska	255.0.0.0	11111111.00000000.00000000.00000000

Aby otrzymać adres sieci muszę wyzerować wszystkie bity w adresie, które odpowiadają zgaszonym bitom w masce podsieci. Tak więc adres sieci dla tego adresu to:

	W postaci dziesiętnej	W postaci bitowej
Adres IP	10.0.0.0	00001010.00000000.00000000.00000000
Maska	255.0.0.0	11111111.00000000.00000000.00000000

W przypadku adresu rozgłoszeniowego muszę zrobić rzecz przeciwną, zapalić wszystkie bity w adresie, które odpowiadają zgaszonym bitom maski:

	W postaci dziesiętnej	W postaci bitowej
Adres IP	10.255.255.255	00001010.11111111.11111111.11111111
Maska	255.0.0.0	11111111.00000000.00000000.00000000

Adres podany w zadaniu to adres komputera. Inny adres komputera z tej sieci to np. 10.42.42.42/8. Adres sieci to 10.0.0.0/8, zaś adres rozgłoszeniowy to 10.255.255.255/8.

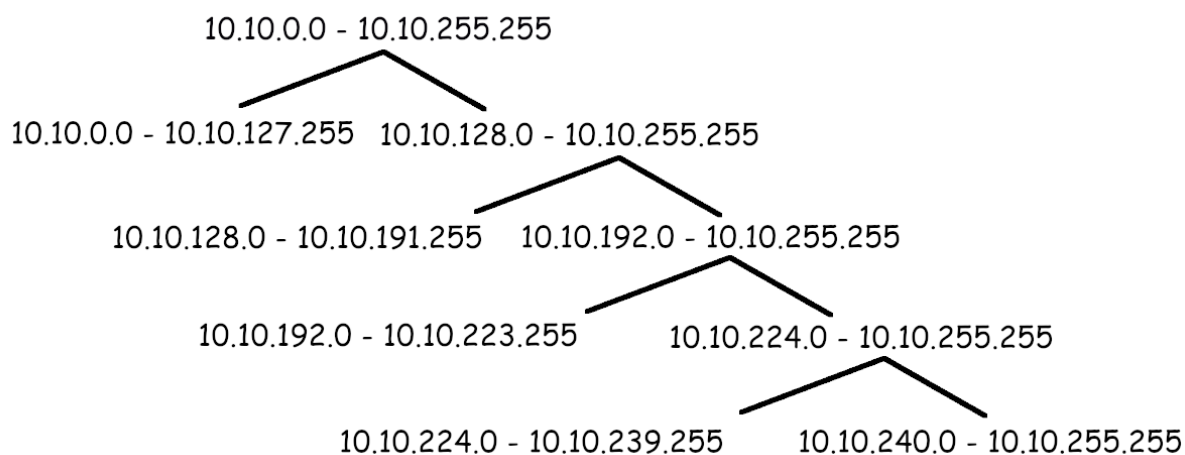
Pozostałe podpunkty są analogiczne, więc podam po prostu odpowiedzi na pytania postawione w poleceniu:

- 156.17.0.0/16
 - Jest to adres sieci
 - Adres broadcast: 156.17.255.255/8
 - Jakiś adres komputera: 156.17.42.42/8
- 99.99.99.99/27
 - Jest to adres komputera
 - Adres sieci: 99.99.99.96/27
 - Jakiś adres komputera: 99.99.99.100/27
 - Adres broadcast: 99.99.99.127/27

- 156.17.64.4/30
 - Jest to adres sieci
 - Jakiś adres komputera: 156.17.64.5/30
 - Adres broadcast: 156.17.64.7/30
- 123.123.123.123/32
 - W przypadku takiej maski jest to pojedynczy adres, nie cała sieć. Nie ma tutaj oddzielnego adresu sieci i rozgłoszeniowego.

2. Podziel sieć 10.10.0.0/16 na 5 rozłącznych podsieci, tak aby każdy z adresów IP z sieci 10.10.0.0/16 był w jednej z tych 5 podsieci. Jak zmieniła się liczba adresów IP możliwych do użycia przy adresowaniu komputerów? Jaki jest minimalny rozmiar podsieci, który możesz uzyskać w ten sposób?

Ta sieć ma zakres adresów: 10.10.0.0 – 10.10.255.255. Poniżej przedstawiam schemat tego jak mogę podzielić tę sieć na 5 podsieci, aby jednocześnie uzyskać jak najmniejszą podsieć (spełniając warunki zadania):



Najmniejsza podsieć ma tutaj 4094 hostów. Uzyskane dzięki temu adresy sieci:

- 10.10.0.0/17
- 10.10.128.0/18
- 10.10.192.0/19
- 10.10.224.0/20
- 10.10.240.0/20

Początkowo w sieci 10.10.0.0/16 mieliśmy 65534 hostów. Przez podział sieci na 5 podsieci utraciliśmy 8 adresów (na rzecz adresów podsieci oraz ich adresów rozgłoszeniowych). W takim razie łączna ilość adresów do przydzielenia hostom (wliczając wszystkie podsieci) to: 65526.

3. Tablica routingu zawiera następujące wpisy (podsieć → dokąd wysłać):

- 0.0.0.0/0 → do routera A
- 10.0.0.0/23 → do routera B
- 10.0.2.0/24 → do routera B
- 10.0.3.0/24 → do routera B
- 10.0.1.0/24 → do routera C
- 10.0.0.128/25 → do routera B
- 10.0.1.8/29 → do routera B
- 10.0.1.16/29 → do routera B
- 10.0.1.24/29 → do routera B

Napisz równoważną tablicę routingu zawierającą jak najmniej wpisów.

Równoważna tablica routingu:

- 0.0.0.0/0 → do routera A
- 10.0.0.0/22 → do routera B
- 10.0.1.0/24 → do routera C
- 10.0.1.8/29 → do routera B
- 10.0.1.16/28 → do routera B

Zostawiamy pierwszy wpis (dzięki temu wszystkie pakiety będą domyślnie wysyłane do A jeśli nie znajdą innej drogi). Następnie łączymy 2, 3 i 4 wpis w jeden: 10.0.0.0/22. Odpowiada on wszystkim adresom tych trzech sieci. Zostawiamy adres 5, ponieważ nigdzie wcześniej nie mieliśmy wpisu o wysłaniu pakietów do routera C. Usuwamy wpis 6 (zawiera się we wpisie 10.0.0.0/22). Zostawiamy wpis 10.0.1.8/29 (inaczej pakiety do tego adresu byłyby wysyłane do routera C, a chcemy do B). Na końcu łączymy ostatnie dwa wpisy w jeden (10.0.1.16/28). To daje nam 5 wpisów w tablicy routingu zamiast początkowych 9.

4. Wykonaj powyższe zadanie dla tablicy

- 0.0.0.0/0 → do routera *A*
- 10.0.0.0/8 → do routera *B*
- 10.3.0.0/24 → do routera *C*
- 10.3.0.32/27 → do routera *B*
- 10.3.0.64/27 → do routera *B*
- 10.3.0.96/27 → do routera *B*

Równoważna tablica routingu:

- 0.0.0.0/0 → do routera *A*
- 10.0.0.0/8 → do routera *B*
- 10.3.0.0/27 → do routera *C*
- 10.3.0.128/25 → do routera *C*

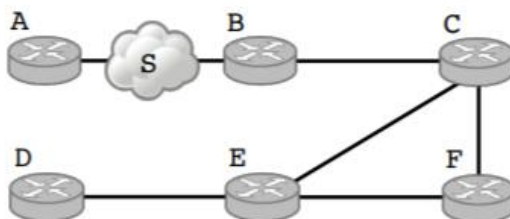
Zostawiamy wpis pierwszy, aby domyślnie pakiety były przesyłane do routera *A* (jeśli nie znajdą innej drogi). Zostawiamy także wpis drugi, ponieważ nigdzie wcześniej nie było wpisu związanego z routerem *B*. Poprzez podział wpisu 3 na dwa wpisy: 10.3.0.0/27 oraz 10.3.0.128/25 możemy pozbyć się trzech ostatnich wpisów (zawierają się we wpisie 10.0.0.0/8). Taki podział uwzględni wszystkie adresy z 10.3.0.0/24, jednocześnie omijając adresy trzy ostatnie adresy we wcześniejszej tablicy routingu. Dzięki temu te trzy adresy będą wysyłane przez wpis 2 w nowej tablicy. Wynikowo otrzymujemy tablicę routingu z 4 wpisami zamiast początkowych 6.

5. Jak uporządkować wpisy w tablicy routingu, żeby zasada najlepszego dopasowania odpowiadała wyborowi „pierwszy pasujący” (tj. przeglądaniu tablicy od początku do końca aż do momentu napotkania dowolnej pasującej reguły)? Odpowiedź uzasadnij formalnie.

Zasada najlepszego dopasowania mówi, że najbardziej pasujący adres w tablicy routingu to ten, który ma najdłuższy wspólny prefix z rozpatrywanym adresem. Innymi słowy jest to adres w tablicy, którego maska ma możliwie najwięcej zapalonych bitów, i którego bity adresu sieci najbardziej pokrywają się z bitami sieci rozpatrywanego adresu.

Jeśli w takim razie potrzebujemy najdłuższego wspólnego prefixu, to najlepiej będzie uporządkować wpisy według długości maski (maski z największą ilością zapalonych bitów na początku).

6. W podanej niżej sieci tablice routingu budowane są za pomocą algorytmu wektora odległości. Pokaż (krok po kroku), jak będzie się to odbywać. W ilu krokach zostanie osiągnięty stan stabilny?



Poniżej rozpiszę tablicę odległości (analogiczną jak ta w prezentacji do wykładu) dla poszczególnych routerów.

KROK 0 – Uzupełnienie wektorów odległości poprzez nadanie sąsiadom długości 1:

	A	B	C	D	E	F
Droga do A	-	1				
Droga do B	1	-	1			
Droga do C		1	-		1	1
Droga do D				-	1	
Droga do E			1	1	-	1
Droga do F			1		1	-
Droga do S	1	1				

KROK 1 – Routery przesyłają swoim sąsiadom swoje tablice routingu. Tak jak na wykładzie, dla czytelności, zakładam, że routery wypisane od lewej do prawej dostają tablice od swoich sąsiadów po kolei. Czyli najpierw A dostanie wpisy od B, potem B od A (z już zaktualizowaną tablicą). W szczególności E dostanie od C informację, że do A może przejść w 2 krokach (ponieważ C ma już w tej chwili zaktualizowaną tablicę), przez co E ma do A drogę długości 3 już w pierwszym kroku.

	A	B	C	D	E	F
Droga do A	-	1	2(via B)		3(via C)	3(via C)
Droga do B	1	-	1		2(via C)	2(via C)
Droga do C	2(via B)	1	-	2(via E)	1	1
Droga do D			2(via E)	-	1	3(via C)
Droga do E		2(via C)	1	1	-	1
Droga do F		2(via C)	1	2(via E)	1	-
Droga do S	1	1	2(via B)		3(via C)	3(via C)

KROK 2 – Routery znowu przesyłają swoim sąsiadom swoje tablice routingu. Zmieniają tylko te pola, które mają puste, lub w których droga jest większa niż uzyskana od sąsiada.

	A	B	C	D	E	F
Droga do A	-	1	2(via B)	4(via E)	3(via C)	3(via C)
Droga do B	1	-	1	3(via E)	2(via C)	2(via C)
Droga do C	2(via B)	1	-	2(via E)	1	1
Droga do D		3(via C)	2(via E)	-	1	3(via C)
Droga do E	3(via B)	2(via C)	1	1	-	1
Droga do F	3(via B)	2(via C)	1	2(via E)	1	-
Droga do S	1	1	2(via B)	4(via E)	3(via C)	3(via C)

KROK 3 – Analogiczny do poprzedniego kroku. Jest to ostatni krok, ponieważ w ewentualnym kolejnym kroku tabela pozostałaby niezmieniona.

	A	B	C	D	E	F
Droga do A	-	1	2(via B)	4(via E)	3(via C)	3(via C)
Droga do B	1	-	1	3(via E)	2(via C)	2(via C)
Droga do C	2(via B)	1	-	2(via E)	1	1
Droga do D	4(via B)	3(via C)	2(via E)	-	1	3(via C)
Droga do E	3(via B)	2(via C)	1	1	-	1
Droga do F	3(via B)	2(via C)	1	2(via E)	1	-
Droga do S	1	1	2(via B)	4(via E)	3(via C)	3(via C)

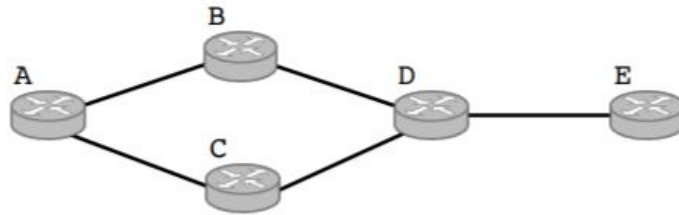
Stan stabilny osiągnięty w 3 krokach (nie licząc kroku 0).

7. Załóżmy, że w powyższej sieci tablice routingu zostały już zbudowane. Co będzie się działo, jeśli zostanie dodane połączenie między routerami A i D?

Tablice routingu w każdym z routerów zostaną zaktualizowane, tak jak w kroku 2. Niektóre z dróg zostaną skrócone (średnica sieci/grafu zmniejsza się poprzez dodanie do niej połączenia między najdalszymi wierzchołkami, więc najdłuższe drogi zostaną skrócone). Tabela z poprzedniego zadania będzie wyglądać następująco (pogrubione zostały zmienione wpisy):

	A	B	C	D	E	F
Droga do A	-	1	2(via B)	1	2(via D)	3(via C)
Droga do B	1	-	1	2(via A)	2(via C)	2(via C)
Droga do C	2(via B)	1	-	2(via E)	1	1
Droga do D	1	2(via A)	2(via E)	-	1	2(via E)
Droga do E	2(via D)	2(via C)	1	1	-	1
Droga do F	3(via B)	2(via C)	1	2(via E)	1	-
Droga do S	1	1	2(via B)	2(via A)	3(via C)	3(via C)

8. W przedstawionej poniżej sieci uszkodzeniu ulega połączenie między routerami *D* i *E*. Załóżmy, że w sieci działa algorytm wektora odległości wykorzystujący technikę zatruwania ścieżki zwrotnej (*poison reverse*). Pokaż — opisując krok po kroku jakie komunikaty są przesyłane między routerami — że może powstać cykl w routingu.



Zatruwanie ścieżki zwrotnej sprawia, że gdy router *D* straci połączenie z routerem *E*, przypisuje on drodze do *E* wartość +nieskończoność i wysyła komunikat o tym do innych routerów. Przykładowy scenariusz powstania pętli:

- *D* mówi *B* i *C*, że do *E* nie ma połączenia (wartość +niesk.)
- *B* wylicza nową trasę przez *A*
 - Mówi *A*, że *E* jest niedostępne
 - Mówi *D*, że ma nową ścieżkę idącą przez *A*
- *C* wlicza nową trasę przez *A*
 - Mówi *A*, że *E* jest niedostępne
 - Mówi *D*, że ma nową ścieżkę idącą przez *A*
- *D* wylicza nową ścieżkę przez *B* lub *C* do routera *E* (CYKL)