

Sieci Komputerowe | Ćwiczenia 1

Cezary Miłek 339746

Zadanie 1

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

- 10.1.2.3/8 - adres komputera:

Maska /8 - 255.0.0.0

- Adres sieci: 10.0.0.0
- Adres rozgłoszeniowy: 10.255.255.255
- Inny adres: 10.1.2.4

- 156.17.0.0/16 - adres sieci:

Maska /16 - 255.255.0.0

- Adres sieci: 156.17.0.0
- Adres rozgłoszeniowy: 156.17.255.255
- Inny adres: 156.17.0.1

- 99.99.99.99/27 - adres komputera:

Maska /27 - 255.255.255.224

- Adres sieci: 99.99.99.96
- Adres rozgłoszeniowy: 99.99.99.127
- Inny adres: 99.99.99.111

- 156.17.64.4/30 - adres sieci:

Maska /30 - 255.255.255.252

- Adres sieci: 156.17.64.4
- Adres rozgłoszeniowy: 156.17.64.7
- Inne adresy: 156.17.64.5 i 156.17.64.6

- 123.123.123.123/32 - maska jest wielkości 32, zatem to jednocześnie adres sieci i rozgłoszeniowy.

Zadanie 2

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?

Propozycja podziału

1. 10.10.0.0/17 - obejmuje adresy 10.10.0.0 - 10.10.127.255 (32 768 adresów)
2. 10.10.128.0/18 - 10.10.128.0 - 10.10.191.255 (16 384 adresy)
3. 10.10.192.0/19 - 10.10.192.0 - 10.10.223.255 (8192 adresy)
4. 10.10.224.0/20 - 10.10.224.0 - 10.10.239.255 (4096 adresów)
5. 10.10.240.0/20 - 10.10.240.0 - 10.10.255.255 (4096 adresów)

Czyli dzielimy sieć na pół, później jedną połówkę na kolejne pół, później znów itd. aż otrzymamy 5 rozłącznych podsieci.

Liczba adresów IP możliwych do użycia zmniejsza się, ponieważ mamy teraz 5 podsieci, z czego każda ma własny adres rozgłoszeniowy oraz adres sieci.

W związku z tym, że początkowo mieliśmy już adres sieci i rozgłoszeniowy, to dla krańcowych podsieci nic się nie zmienia.

Oznacza to, że tracimy w sumie 8 adresów (10 - 2 krańcowe = 8).

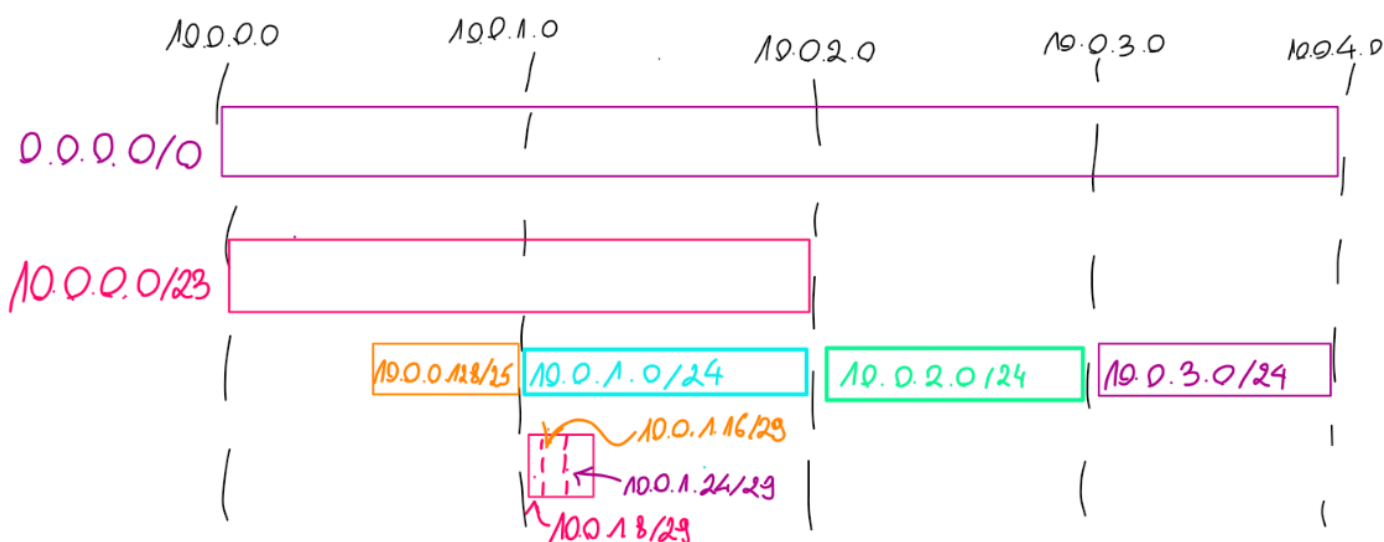
W tym przypadku, przy założeniu, że całość ma się składać dokładnie z 5 rozłącznych podsieci najmniejszą podsiecią jest /20 o rozmiarze 4096.

Zadanie 3

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.



Zauważmy, że:

1. 10.0.0.0/23 (maska 255.255.254.0) obejmuje zakres 10.0.0.0 – 10.0.1.255.
2. Mamy tam jednak wpis 10.0.1.0/24 → C, który jest **bardziej szczegółowy** (prefiks /24) i „przekierowuje” ruch do C dla całego zakresu 10.0.1.x.
3. Następnie w tym 10.0.1.0/24 mamy jeszcze bardziej szczegółowe podsieci /29, które z kolei wracają do B:
 - 10.0.1.8/29 → B
 - 10.0.1.16/29 → B
 - 10.0.1.24/29 → B

Te trzy bloki obejmują łącznie adresy 10.0.1.8 – 10.0.1.31. Można zauważyć, że

10.0.1.16/29 (16–23) i 10.0.1.24/29 (24–31) da się scalać w 10.0.1.16/28 (16–31).

4. 10.0.0.128/25 → B leży w obrębie 10.0.0.0/23 → B i **wskazuje na tego samego B**, więc jest zbędne (bardziej szczegółowa trasa nic nie zmienia, skoro obie mówią „do B”).
5. 10.0.2.0/24 → B i 10.0.3.0/24 → B można scalać w 10.0.2.0/23 → B (obejmuje 10.0.2.0 – 10.0.3.255).

Możemy teraz jeszcze lepiej „zagregować” wpisy 10.0.0.0/23 i 10.0.2.0/23, łącząc je w 10.0.0.0/22 → B (obejmuje 10.0.0.0 – 10.0.3.255). Potem na bardziej szczegółowym poziomie i tak mamy 10.0.1.0/24 → C, a w nim wyjątki do B.

Nowa tablica routingu:

- ▶ 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
- Handwritten notes:*
- Orange curly brace grouping 10.0.2.0/24 and 10.0.3.0/24, pointing to 10.0.2.0/23.
 - Purple curly brace grouping 10.0.0.0/23 and 10.0.2.0/23, pointing to 10.0.0.0/22.
 - Purple arrow pointing from the crossed-out line to the text „zbędne” (redundant).
 - Purple arrow pointing from the text „w tym” (in this) to the 10.0.0.0/22 aggregation.
 - Pink curly brace grouping 10.0.1.8/29, 10.0.1.16/29, and 10.0.1.24/29, pointing to 10.0.1.16/28.

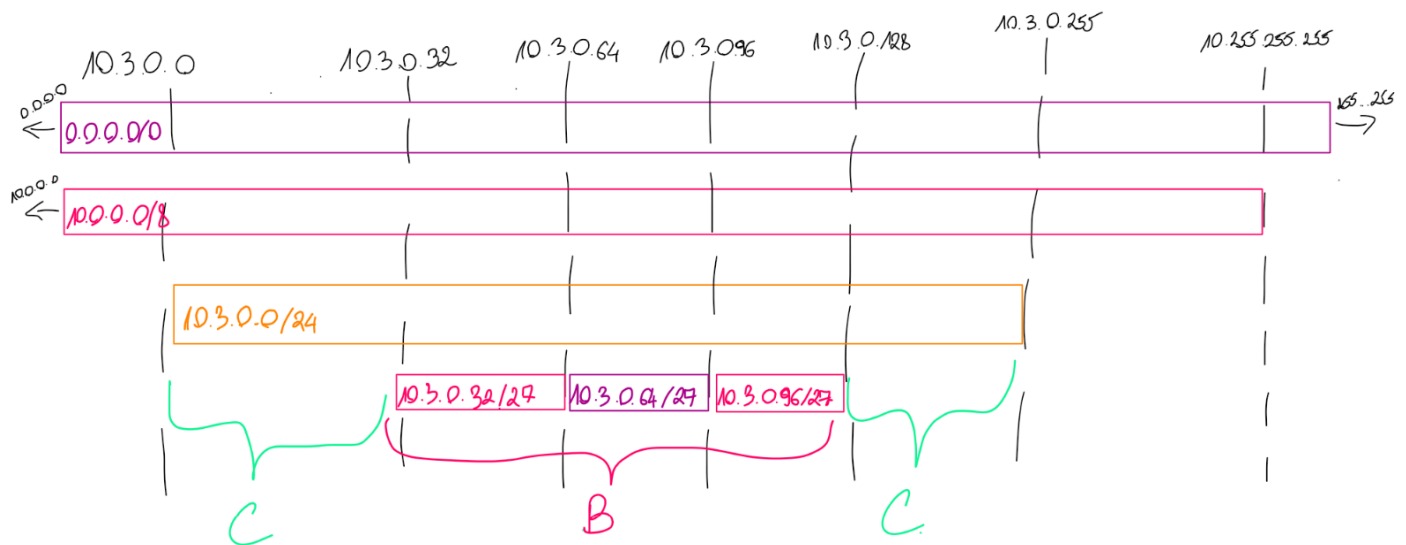
1. 0.0.0.0/0 → A
(domyślna trasa)
2. 10.0.0.0/22 → B
(obejmuje 10.0.0.x, 10.0.1.x, 10.0.2.x, 10.0.3.x)
3. 10.0.1.0/24 → C
(bardziej szczegółowy zakres 10.0.1.x do C)
4. 10.0.1.8/29 → B
5. 10.0.1.16/28 → B
(zamiast dwóch wpisów 16/29 i 24/29, mamy jeden /28)

Dzięki temu mamy łącznie **5 wpisów**.

Zadanie 4

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



Zauważmy:

- Na routerze (C) obsługujemy zakres, w którym zawiera się część adresów obsługiwanych przez router (B).
 - Mając to na uwadze moglibyśmy rozbić 10.3.0.0/24 na dwie części obsługujące (C) i zostawić część routingu (B) "z góry".

Nowa tablica routingu:

1. 0.0.0.0/0 → A
2. 10.0.0.0/8 → B
3. 10.3.0.0/27 → C
4. 10.3.0.128/25 → C

Zadanie 5

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.

Należy posortować wpisy od najbardziej szczegółowego do najbardziej ogólnego, czyli malejąco po długości maski.

W ten sposób, jeśli dany adres IP pasuje do kilku wpisów to pierwszy (od góry) pasujący będzie zarazem tym o najdłuższym prefiksie.

Uzasadnienie:

Założmy, że wpisy są uporządkowane od najdłuższych do najkrótszych masek.

Jeżeli wpis (A) ma dłuższy prefix niż wpis (B) , to (A) jest podsiecią zawartą w (B) .

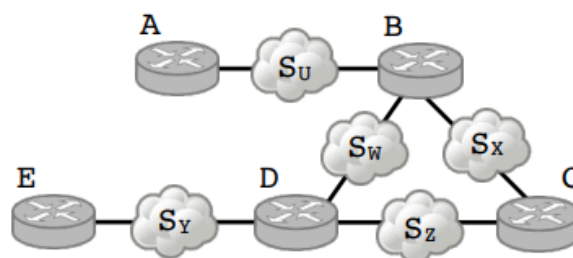
Zatem zawsze, gdy dany adres pasuje do (A) , to pasuje też do (B) , ale **nie odwrotnie**.

Wynika to z tego, że skoro adres pasuje do (A) na (n) bitach, to ten sam adres nie może pasować do (B) na więcej niż (n) bitach (bo ma krótszy prefix).

Umieszczenie (A) wyżej sprawia, że jest on wybrany przed (B) , co jest równoważne podejściu "pierwszy pasujący".

Zadanie 6

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?



Tablica odległości od (X) do (X) :

Krok 0

Router	od A	od B	od C	od D	od E
do \((S_U)\)	1	1			
do \((S_W)\)		1		1	
do \((S_X)\)		1	1		
do \((S_Y)\)				1	1
do \((S_Z)\)			1	1	

Krok 1

Router	od A	od B	od C	od D	od E
do \((S_U)\)	1	1	2 (via B)	2 (via B)	
do \((S_W)\)	2 (via B)	1	2 (via B)	1	2 (via D)
do \((S_X)\)	2 (via B)	1	1	2 (via B)	
do \((S_Y)\)		2 (via D)	2 (via D)	1	1
do \((S_Z)\)		2 (via C)	1	1	2 (via D)

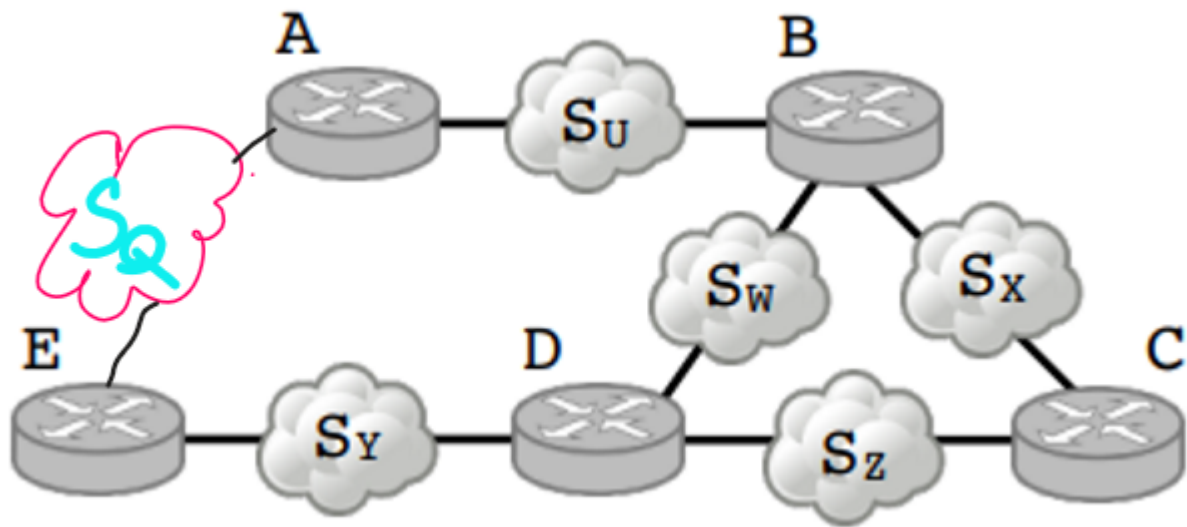
Krok 2

Router	od A	od B	od C	od D	od E
do \((S_U)\)	1	1	2 (via B)	2 (via B)	3 (via D)
do \((S_W)\)	2 (via B)	1	2 (via B)	1	2 (via D)
do \((S_X)\)	2 (via B)	1	1	2 (via B)	3 (via D)
do \((S_Y)\)	3 (via B)	2 (via D)	2 (via D)	1	1
do \((S_Z)\)	3 (via B)	2 (via C)	1	1	2 (via D)

Stan stabilny został osiągnięty w \((2)\) krokach.

Zadanie 7

7. Załóżmy, że w powyższej sieci tablice routingu zostały już zbudowane. Co będzie się działo (krok po kroku), jeśli zostanie dodana sieć S_Q łącząca routery A i E ?



Krok 0

Router	od A	od B	od C	od D	od E
do \((S_U)\)	1	1	2 (via B)	2 (via B)	3 (via D)
do \((S_W)\)	2 (via B)	1	2 (via B)	1	2 (via D)
do \((S_X)\)	2 (via B)	1	1	2 (via B)	3 (via D)
do \((S_Y)\)	3 (via B)	2 (via D)	2 (via D)	1	1
do \((S_Z)\)	3 (via B)	2 (via C)	1	1	2 (via D)
do \((S_Q)\)	1				1

Krok 1

Router	od A	od B	od C	od D	od E
do \((S_U)\)	1	1	2 (via B)	2 (via B)	3 (via D)
do \((S_W)\)	2 (via B)	1	2 (via B)	1	2 (via D)
do \((S_X)\)	2 (via B)	1	1	2 (via B)	3 (via D)
do \((S_Y)\)	3 (via B)	2 (via D)	2 (via D)	1	1
do \((S_Z)\)	3 (via B)	2 (via C)	1	1	2 (via D)
do \((S_Q)\)	1	2 (via A)		2 (via E)	1

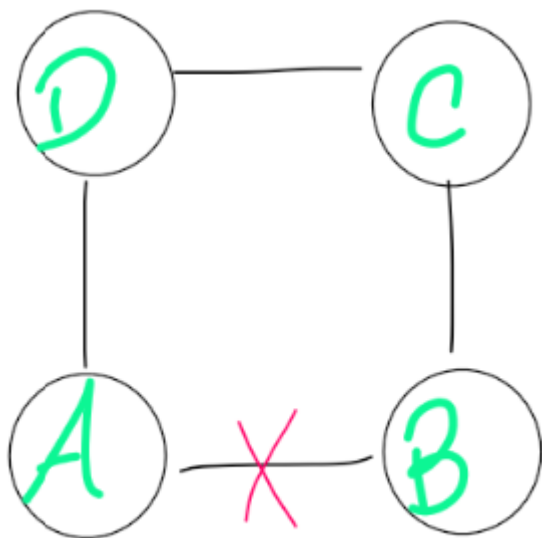
Krok 2

Router	od A	od B	od C	od D	od E
do $\backslash(S_U\backslash)$	1	1	2 (via B)	2 (via B)	3 (via D)
do $\backslash(S_W\backslash)$	2 (via B)	1	2 (via B)	1	2 (via D)
do $\backslash(S_X\backslash)$	2 (via B)	1	1	2 (via B)	3 (via D)
do $\backslash(S_Y\backslash)$	3 (via B)	2 (via D)	2 (via D)	1	1
do $\backslash(S_Z\backslash)$	3 (via B)	2 (via C)	1	1	2 (via D)
do $\backslash(S_Q\backslash)$	1	2 (via A)	3 (via B)	2 (via E)	1

Stan stabilny został osiągnięty w $\backslash(2\backslash)$ krokach.

Zadanie 9

9. Pokaż, że przy wykorzystaniu algorytmu stanu łączy też może powstać cykl w routingu. W tym celu skonstruuj sieć z dwoma wyróżnionymi, sąsiadującymi ze sobą routerami A i B . Załóż, że wszystkie routery znają graf całej sieci. W pewnym momencie łącze między A i B ulega awarii, o czym A i B od razu się dowiadują. Zalewają one sieć odpowiednią aktualizacją. Pokaż, że w okresie propagowania tej aktualizacji (kiedy dotarła ona już do części routerów a do części nie) może powstać cykl w routingu.



Założmy, że doszło już do awarii między $\backslash(A\backslash)$ i $\backslash(B\backslash)$. Dowiadują się o tym od razu i zaczynają aktualizować resztę routerów.

Założmy, że najszybsza droga przesłania pakietu z $\backslash(D\backslash)$ do $\backslash(B\backslash)$ wiodła przez $\backslash(A\backslash)$.

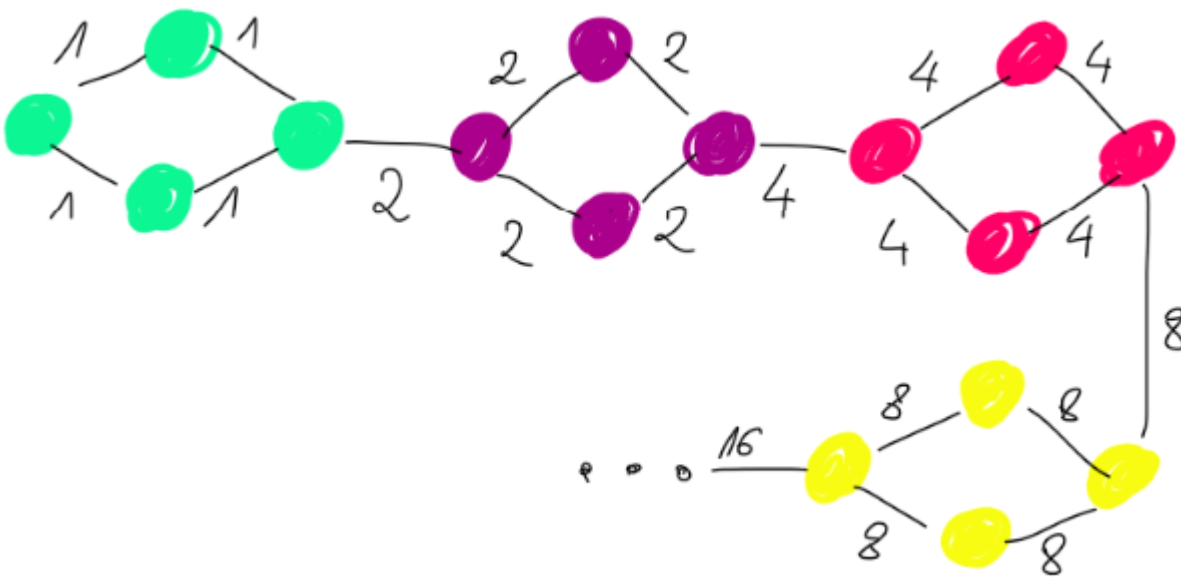
Jeśli (D) chce wysłać coś do (B) , to pakiet dostarczany jest do (A) i stąd (A) powinno przesłać go do (B) . Jednak jest awaria w tym połączeniu - (A) zwraca pakiet do (D) .

(D) jest jednak niezaktualizowane, w związku z czym dalej próbuje wysłać pakiet standardową wcześniej dla siebie drogą, czyli przez (A) . Powstaje cykl.

Pięteczka jest odbijana cały czas w tym cyklu, aż aktualizacja nie dotrze do odpowiednich routerów.

Zadanie 10

10. Załóżmy, że sieć składa się z łączy jednokierunkowych (tj. łączy w sieci tworzą graf skierowany) i nie zawiera cykli. Rozważmy niekontrolowany algorytm „zalewający” sieć jakimś komunikatem: komunikat zostaje wysłany początkowo przez pewien router; każdy router, który dostanie dany komunikat przesyła go dalej wszystkimi wychodzącymi z niego krawędziami. Pokaż, że istnieją takie sieci z n routerami, w których przesyłanie informacji zakończy się po czasie $2^{\Omega(n)}$. Zakładamy, że przez jedno łącze można przesłać tylko jeden komunikat naraz, a przesłanie go trwa jednostkę czasu.



Każdy segment zwiększa podwójnie czas zakończenia, ponieważ jedno łącze może przesłać tylko jeden komunikat naraz, a przesłanie trwa jednostkę czasu.

W związku z tym tworzy się kolejka przed każdym segmentem, z każdym nowym segmentem o dwa razy dłuższa.

Oznacza to, że na końcowym segmencie będziemy mieli $2^{\Omega(n)}$ jako końcowy czas przesłania informacji.