

**DETERMINISTYCZNY BROADCAST W SIECIACH
RADIOWYCH**

ŁUKASZ JEŹ

SPIS TREŚCI

1. Wstęp	4
2. Modele	4
2.1. Model bez detekcji kolizji	4
2.2. Model z detekcją kolizji	6
3. Prosty protokół i jego rozwinięcia	9
3.1. Protokół ROUND-ROBIN	9
3.2. Pewien szczególny przypadek	10
3.3. Rodziny selektywne, silnie selektywne oraz selektory	11
3.4. Niekonstruktywne ograniczenia górne na rozmiary rodzin selektywnych i selektorów	12
4. Nietrywialne górne ograniczenia na czas rozgłaszania	16
4.1. Protokół zależny od n , D i d	16
4.2. Szybki protokół rozgłaszania	20
4.3. Przeplatanie protokołów i szybsza terminacja	22
5. Dolne ograniczenia na rozmiar rodzin selektywnych, rodzin silnie selektywnych i selektorów	23
5.1. Dolne ograniczenia na czas rozgłaszania	25
5.2. Protokół rozgłaszania w grafach warstwowych o optymalnym czasie powiadomienia	27
6. Kilka słów o uogólnionych selektorach	30
6.1. Uogólnione selektory oraz ich związki z poprzednimi strukturami	30
6.2. Ograniczenia na rozmiar uogólnionych selektorów	31
6.3. Ograniczenia na rozmiar selektorów	31
7. Konstrukcja selektorów	32
7.1. Wstęp do konstrukcji	32
7.2. Właściwa konstrukcja	33
8. Grafy nieskierowane	35
8.1. Liniowy czas przy spontanicznej komunikacji	35
8.2. Symulowanie detekcji kolizji	36
8.3. Pełne nieskierowane grafy warstwowe oraz ograniczenia dolne	41
8.4. Grafy nieskierowane w modelu z detekcją kolizji	43
9. Niemożliwość „świadomego rozgłaszania”	48
10. Problemy otwarte	51
Literatura	52

1. WSTĘP

Rozważamy problem deterministycznego rozgłaszania (*broadcast*) w sieciach radiowych oraz jego związek z pewnymi konstrukcjami kombinatorycznymi: rodzinami selektywnymi, rodzinami silnie selektywnymi oraz selektorami. Dzięki tym konstrukcjom można nie tylko uzyskać efektywne protokoły rozgłaszania w sieciach radiowych, ale także ograniczenia dolne dla tego problemu.

Sieci radiowe różnego rodzaju, np. telefonia komórkowa czy bezprzewodowy Internet są już intensywnie używane. Protokoły komunikacji w takich sieciach powinny zakładać możliwie mało o topologii sieci, gdyż ulega ona ciągłym zmianom. Rozważamy protokoły z minimalną wiedzą konieczną do przeprowadzenia rozgłaszania — każdy wierzchołek sieci zna wyłącznie swój identyfikator.

Dokładnie jeden węzeł sieci, źródło, posiada *wiadomość początkową* m , którą ma następnie przekazać pozostałym węzłom w możliwie krótkim czasie. Zadanie to utrudnia interferencja — jeśli do pewnego węzła nadaje jednocześnie kilka innych, wzajemnie się zagłuszają i węzeł ten w rezultacie nie odbiera żadnego sygnału. Czasem zakłada się możliwość *detekcji kolizji*: wtedy węzeł potrafi stwierdzić, że kilka węzłów próbowało jednocześnie przekazać mu pewne komunikaty. Zajmujemy się głównie modelem bez detekcji kolizji. Podamy jednak kilka wyników dla modelu z detekcją kolizji, które ilustrują, jak pomaga ona w rozgłaszaniu.

Większość omawianych protokołów charakteryzuje się *brakiem spontanicznej komunikacji*, co znaczy, że żaden węzeł poza źródłem nie nadaje komunikatów, dopóki sam jakiegось nie odbierze. Węzeł, który nie otrzymał żadnego sygnału i w konsekwencji nie może nadawać, to *węzeł uśpiony*. Gdy odbiera sygnał po raz pierwszy, *budzi się* i jest od tej pory *węzłem aktywnym*. Źródło jest węzłem aktywnym od początku wykonywania protokołu.

Brak spontanicznej komunikacji to oczywiście ograniczenie — nie pozwala węzłom zbierać informacji o swoim otoczeniu, które mogłyby wykorzystać do szybszego przekazania wiadomości początkowej, gdy ją wreszcie otrzymają. Za to protokoły bez spontanicznej komunikacji są prostsze i mogą być użyte również w problemie *budzenia w sieci*.

Problem budzenia jest zbliżony do problemu rozgłaszania. W problemie budzenia w sieci początkowo źródło jest aktywne a pozostałe węzły uśpione. Bez spontanicznej komunikacji należy obudzić wszystkie węzły. Podobnie jak w problemie rozgłaszania, każdy z węzłów zna tylko swój identyfikator. Nadal dochodzi do interferencji, zaś detekcja kolizji jest niedostępna (problem z detekcją kolizji trywializuje się). Związek powyższych problemów jest oczywisty: protokoły rozgłaszania bez spontanicznej komunikacji służyć mogą jako protokoły budzenia — wystarczy zignorować wiadomość początkową i przekazywać dowolny sygnał.

Protokoły budzenia, dzięki prostocie znajdują liczne praktyczne zastosowania, choćby w wykrywaniu pożarów lasów: w lesie rozrzuca się proste i tanie urządzenia, mogące odbierać sygnały radiowe oraz nadawać słabe sygnały na krótkie odległości. Gdy urządzenie wykryje wysoką temperaturę, budzi się i rozpoczyna protokół. Jeśli urządzenia są rozrzucone odpowiednio gęsto, przekazują sygnał między sobą aż odbierze go znajdujący się na obrzeżach lasu nadajnik o dużej mocy. Ten może przekazać informację o pożarze na dużą odległość.

2. MODELE

2.1. Model bez detekcji kolizji. Sieć radiową reprezentujemy jako graf skierowany $G = (V, E, s)$. $s \in V$ oznacza źródło, z którego osiągalny jest każdy wierzchołek. Przyjmujemy, że $V = [n] = \{1, 2, \dots, n\}$ dla pewnego (nieznanego) n . Jeśli nie będzie to rodzić niejasności, będziemy utożsamiać wierzchołek z jego identyfikatorem. W szczególności zakładamy, że identyfikatory to liczby naturalne od 1 do

n . Zadaniem jest opracowanie protokołu, umożliwiającego przekazanie wiadomości początkowej m , którą zna źródło s , do pozostałych wierzchołków. Wierzchołki grafu czasem będziemy nazywać węzłami, by podkreślić, że chodzi o elementy sieci.

Graf G oraz jego parametry nie są znane, tj. protokół musi działać poprawnie dla każdego grafu. Węzły mają dostęp do globalnego zegara odmierzającego dyskretny czas od 0. Odcinek czasu od t do $t+1$ nazywamy rundą t . Protokół specyfikuje dla każdego węzła v i każdej rundy t , czy w rundzie t węzeł v nasłuchuje, czy nadaje komunikat, również specyfikowany przez protokół. Akcja każdego węzła v zależy wyłącznie od

- jego identyfikatora v ,
- chwili t ,
- historii węzła v , tj. zapisu odebranych przez niego komunikatów w rundach od 0 do t ,

przy czym na ogół zakładamy brak spontanicznej komunikacji. Oznacza to, że węzeł o pustej historii nie nadaje komunikatów. Dla uproszczenia przyjmiemy, że protokół rozpoczyna się w rundzie 1, za to źródło otrzymuje wiadomość m w rundzie 0.

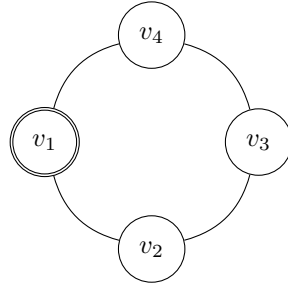
Zakładamy, że każdy węzeł w ciągu rundy zdąży obliczyć, jaką akcję ma podjąć oraz wykonać ją. Jeśli w rundzie t węzeł v nasłuchuje, odbiera komunikat c wtedy i tylko wtedy, gdy c jest komunikatem nadawanym przez jedyne poprzednika v nadającego w rundzie t . Zakładamy, że sygnały nadawane przez węzły są na tyle krótkie, że można je przesłać w jednej rundzie. Nie zawsze jest to uzasadnione — prezentujemy m.in. protokoły, w których węzły nadają swoje identyfikatory, lub inne komunikaty zależne od wielkości sieci.

Łatwo zauważyć, że przy braku spontanicznej komunikacji globalny zegar nie jest potrzebny — wystarczy globalny metronom! Węzłom wystarczy wspólne odmierzenie rund, niekoniecznie muszą znać numery tych rund. Węzeł v musi poznać numer rundy dopiero gdy ma nadać jakiś komunikat. Nim to nastąpi, sam odbierze jakiś komunikat. W tym komunikacie może być zawarty numer rundy: źródło rozpoczyna odliczanie od 1 i dokleja numer rundy do każdego komunikatu, który wysła. Węzeł, który otrzyma komunikat z numerem rundy t , wie, że otrzymał go w rundzie $t+1$. Odtąd może sam liczyć rundy i doklejać ich numery do nadawanych komunikatów.

Najwcześniejszą rundę t , w której wszystkie węzły znają wiadomość m nazywamy *czasem powiadomienia protokołu*. Z kolei *czasem terminacji* nazywać będziemy najwcześniejszą rundę, poczynając od której żaden z węzłów nie nadaje komunikatów. Implicite zakładamy, że węzeł, który w rundzie t nie nadaje żadnego komunikatu, nasłuchuje. Czas terminacji został wprowadzony w [CMS01], jednak większość autorów nie dbała o terminację. Uzupełniamy twierdzenia dotyczące czasów powiadomienia różnych protokołów o analizę czasów terminacji.

W pseudokodach protokołów będziemy zwykle określać, które węzły nadają jakie komunikaty w danej rundzie. Prawdziwy kod powinien być pisany dla węzła, tj. specyfikować dla niego, jaką akcję ma wykonać. W związku z tym zaznaczamy, że słowo kluczowe *foreach* rozumiemy jako równoległe wykonanie operacji przez wszystkie węzły z danego zbioru a nie wykonywanie tychże operacji sekwencyjnie przez kolejne węzły.

2.1.1. Schemat podwajania. Dla prostoty opisywane protokoły będą korzystały z wartości $n = |V|$, mimo że nie jest ona znana. Oczywiście poprawny protokół dla n węzłów, będzie dobry dla dowolnego $m \leq n$. Dlatego implicite stosujemy następujący *schemat podwajania*: wykonujemy kolejno protokół dla $n = 2^0, 2^1, 2^2, \dots$. Dowolny ciąg rosnący jest dobry, ale ten gwarantuje, że zachowujemy asymptotyczną wielkość czasu powiadomienia. Niech wynosi on $T(n)$ dla sieci o n węzłach.

RYSUNEK 1. Cykl C_4 z wyróżnionym źródłem.

Wtedy po wykonaniu protokołu dla 2^k takiego, że $2^{k-1} < n \leq 2^k$, wszystkie węzły będą znały m . Oczywiście $2^k < 2n$. Jeśli więc $T(n)$ jest wypukłą funkcją n , to czas powiadomienia wynosi $T'(n) = \sum_{j=0}^k T(2^j) = \mathcal{O}(T(2^k)) = \mathcal{O}(T(n))$.

Pokażemy, że czas powiadomienia każdego protokołu poprawnego dla dowolnego grafu musi wynosić $\Omega(n \log n)$. Jest to wypukła funkcja n , czyli czas powiadomienia zachowuje swą asymptotyczną wielkość. Podobnie będzie w szybszych protokołach dla grafów szczególnych oraz wtedy, gdy czas powiadomienia wyrazimy jako funkcję większej liczby zmiennych, np. d — maksymalnego stopnia wejściowego grafu i D — maksymalnej odległości wierzchołka od źródła. O ewentualnych odstępstwach od tej reguły będziemy wyraźnie mówić.

Zauważmy, że uzyskany w ten sposób protokół nie posiada gwarancji na czas terminacji: nawet gdy wszystkie węzły poznały już m , będziemy za pewien czas próbować większego n . Wobec tego protokół w ogóle nie terminuje. O terminację i jej czas będziemy dodatkowo dbali.

2.1.2. Czemu identyfikatory węzłów są konieczne. Pokażemy teraz, że unikalne identyfikatory węzłów są konieczne, by przeprowadzić deterministyczny broadcast. Wykażemy to nawet dla szczególnego przypadku grafów nieskierowanych. Zajmujemy się protokołami deterministycznymi. Brak dostępu do bitów losowych sprawia, że pewne (różne od źródła) „symetryczne” węzły zawsze zachowują się w ten sam sposób. Można ich użyć jako „blokady”, nie przepuszczającej sygnałów z jednej części grafu do drugiej — będą albo jednocześnie nasłuchiwać, albo nadawać (te same sygnały!), wzajemnie się zagłuszając. Najprostszym tego typu przykładem jest graf C_4 , tj. cykl o czterech wierzchołkach, widoczny na Rysunku 1. Ponumerujemy jego wierzchołki, zaczynając od źródła, tj. źródłem jest v_1 , zaś dalej w ustalonym kierunku mamy v_2, v_3 i v_4 , sąsiadujący z v_1 . Parą wierzchołków o identycznych zbiorach sąsiadów są v_2 i v_4 . Odcinają one v_4 od źródła, przez co v_4 nie może poznać m . Warto zauważyć, że podany warunek powodujący identyczne zachowanie węzłów jest silny — w istocie wystarczy, by ich zbiory sąsiadów zachowywały się identycznie, niekoniecznie muszą być identyczne.

2.2. Model z detekcją kolizji. Rozważany, choć rzadziej, jest również model, w którym węzeł v próbujący odebrać sygnał w rundzie t , w której nadaje co najmniej dwóch jego poprzedników, odbiera specjalny sygnał μ . Oznacza on, że doszło do interferencji, tj. v dowiadyuje się, że co najmniej dwa węzły próbowały przekazać mu pewne komunikaty. Jest to jedyna informacja, jaką uzyskuje v : nie poznaje liczby nadających poprzedników, identyfikatora żadnego z nich, ani żadnego z nadawanych komunikatów. Dla powyższego modelu z detekcją kolizji podajemy dwa przykłady z [CGGPR00], które ilustrują siłę detekcji kolizji.

W obu przykładach protokoły rozróżniają tylko, czy dany węzeł odebrał cokolwiek, tj. μ lub pewien komunikat c , czy nie odebrał nic. Z tego piszemy, że węzeł

nadaje μ , gdy rolę odgrywa jedynie fakt, że węzeł nadaje a nie jest ważny sam komunikat. Jest to wygodna konwencja, bo węzeł v odbiera w rundzie t sygnał μ o ile tylko w rundzie t nadawaje co najmniej jeden poprzednik v . Po przyjęciu tej konwencji łatwo stwierdzić, że problem budzenia w sieci radiowej przy dostępnej detekcji kolizji się trywializuje — istnieje oczywisty protokół budzący wszystkie węzły i terminujący w czasie D .

Pewien kłopot stanowi zdefiniowanie braku spontanicznej komunikacji w modelu z detekcją kolizji. Przyjrzyjmy się bowiem protokołom bez spontanicznej komunikacji i bez detekcji kolizji. Doklejmy m (z odpowiednim znacznikiem) do każdego komunikatu, który go nie zawiera. Teraz każdy obudzony węzeł zna m i w definicji możemy przyjąć równoważnie, że nadawać mogą jedynie węzły, które odebrały m . Za to gdy dopuszczamy detekcję kolizji, węzeł może odebrać dwa różne typy sygnałów: komunikat nadany przez poprzednika (o którym możemy założyć, że zawiera m), lub sygnał interferencji μ , niosący dużo mniej informacji. Nie jest jasne, kiedy uznać węzeł za obudzony: po otrzymaniu m czy już po usłyszeniu μ . Tym bardziej, że, jak pokażemy, węzły mogą „odczytać” m ze swojej historii, będącej sekwencją ciszy i sygnałów μ w kolejnych rundach — a więc sekwencją bez „pełnowartościowych” komunikatów.

2.2.1. *Grafy silnie spójne.* Pokażemy, że jeśli sieć jest grafem silnie spójnym, to w czasie $\mathcal{O}(n)$ węzły sieci mogą poznać n z dokładnością do stałego czynnika. Ścisiej, są w stanie znaleźć k naturalne takie, że $2^{k-1} < n \leq 2^k$. Potem wystarczy wykonać protokół rozgłaszania dla $n' = 2^k$. Przypominamy, że stosowany w modelu bez detekcji kolizji schemat podwajania nie pozwala wprost na przerwanie wykonania protokołu rozgłaszania. Stosując technikę opisaną poniżej, dostajemy protokół, który ma tę samą gwarancję na czas terminacji co na czas powiadomienia.

Protokół BOUND służy do wyznaczenia n' jak wyżej i składa się z kolejnych faz numerowanych od 0, gdzie faza k składa się z $2^k + 1$ rund numerowanych od 1. Wyjątkiem jest faza 0, mająca tylko jedną rundę. Po k -tej fazie węzeł v będziemy nazywać *aktywnym*, jeśli w tej fazie nadał μ . W jedynej rundzie fazy 0 wszystkie węzły nadają μ . W pierwszej rundzie fazy k , μ nadają wszystkie aktywne węzły o identyfikatorach większych od 2^k . W rundzie i ($i > 1$) fazy k sygnał μ nadają wszystkie węzły, które po raz pierwszy w fazie k odebrały μ w rundzie $i - 1$. Każdy węzeł wyznacza $n' = 2^k$, gdzie k jest numerem fazy, po której węzeł ten się dezaktywował.

Protokół BOUND spełnia następujący niezmiennik:

Niezmiennik 1. *Po każdej z faz, albo wszystkie węzły są aktywne, albo wszystkie są nieaktywne.*

Dowód. Indukcja względem liczby faz oraz liczby rund w każdej z faz. Opiszemy ją w skrócie. W fazie k o węzłach z identyfikatorami ze zbioru $[2^k]$ powiemy, że mają *niskie identyfikatory* a o pozostałych, że mają *wysokie identyfikatory*

W pierwszej rundzie fazy k , gdzie $n > 2^k$, tj. $k < \lceil \log n \rceil$, niektóre wierzchołki o niskich identyfikatorach odbierają sygnał μ nadany przez węzły o wysokich identyfikatorach, bo G jest silnie spójny. W następnej rundzie same będą nadawać, a ich sygnał dotrze do kolejnych węzłów. Nie licząc pierwszej rundy, rund jest tyle co węzłów o niskich identyfikatorach. Czyli wystarczająco dużo, by każdy węzeł o niskim identyfikatorze odebrał oraz nadał sygnał μ . Zatem wszystkie węzły pozostają aktywne.

Za to w fazie $\lceil \log n \rceil$ żaden węzeł nie nadaje, więc po zakończeniu tej fazy wszystkie węzły się dezaktywują. \square

Z niezmiennika wprost wynika poniższe twierdzenie:

```

Dane niejawne:  $G(V, E)$ , silnie spójny
Dane jawne:  $s$ 
Wynik:  $n': \frac{n'}{2} < n \leq n'$ 

foreach  $v$  do /* faza 0 */
     $v$  nadaje  $\mu$ ;
for  $k = 1, 2, \dots$  do /* kolejne fazy */
    foreach  $v > 2^k$  do /* runda 1 fazy  $k$  */
         $v$  nadaje  $\mu$ ;
        for  $i = 2, 3, \dots, 2^k + 1$  do /* kolejne rundy fazy  $k$  */
            foreach  $v$  do
                if runda  $i - 1$  była pierwszą rundą fazy  $k$ , w której  $v$  odebrał  $\mu$  then
                     $v$  nadaje  $\mu$ ;
            foreach  $v$  do
                if  $v$  nie nadał  $\mu$  w tej fazie then
                     $v$  stwierdza, że  $n' = 2^k$  i przerywa protokół

```

Protokół 1: Protokół BOUND

Twierdzenie 1. Protokół BOUND terminuje w czasie $\mathcal{O}(n)$ dla dowolnego grafu silnie spójnego o n wierzchołkach oraz wyznacza n' takie, że $\frac{n'}{2} < n \leq n'$. Co więcej, n' znane jest każdemu węzłowi sieci i wszystkie węzły poznają je w tym samym momencie.

2.2.2. *Znacząca cisza: kodowanie wiadomości w szumie.* Przypuśćmy, że węzeł v wie, że w danej rundzie pewien podzbiór poprzedników v chce przesłać do niego bit b , wszystkie ten sam. Zamiast się zagłuszać, mogą współpracować! Jeśli $b = 0$, nie nadają nic, jeśli zaś $b = 1$, nadają μ . v umie odtworzyć wartość b : $b = 0$ wtedy i tylko wtedy, gdy nie odebrał nic. To spostrzeżenie jest podstawą Protokołu ENCODED-BROADCAST, w którym przekazywane są kolejne bity $m = b_1, b_2, \dots, b_r$. Wystarczy zadbać o to, by węzły wiedziały, kiedy m jest do nich w ten sposób przesyłane, bo w przeciwnym razie źle interpretowałyby ciszę.

Protokół składa się z kolejnych faz. W pierwszej s nadaje m . Kolejne fazy mają po $2r + 4$ rund. W fazie k ($k > 1$), aktywne są węzły, które odebrały μ po raz pierwszy w fazie $k - 1$. Zachowany będzie następujący niezmiennik: węzły aktywne znają m . W każdej z faz nadają wyłącznie aktywne węzły. W pierwszych i ostatnich dwóch rundach, aktywne węzły nadają μ . W rundach $2i + 1, 2i + 2$, postępują zależnie od wartości b_i : jeśli $b_i = 0$, to w obu nie nadają, zaś gdy $b_i = 1$, w pierwszej z nich nie nadają, zaś w drugiej nadają μ . Następnicy węzłów aktywnych rozpoznają początek i koniec transmisji jako podwójne μ , zaś z transmisji pomiędzy nimi odtwarzają m (ignorując sygnały nieparzyste, a parzyste traktując jak to opisano wcześniej). Każdy węzeł zaraz po tym, jak pozna m , przekazuje je dalej, a następnie się dezaktywuje — oznacza to, że w czasie $\mathcal{O}(|m|D)$ nie tylko wszystkie węzły poznają m , ale także protokół terminuje. Zaznaczmy, że przy rozsądnym założeniu, że w jednej rundzie przesłać tylko stałą liczbę bitów, wynik ten jest optymalny, choć niepraktyczny.

Protokół ENCODED-BROADCAST posiada dwie ciekawe własności:

- (1) nie korzysta z identyfikatorów węzłów,

(2) akcje węzłów zależą od m .

Pierwsza oznacza, że w modelu z detekcją kolizji identyfikatory nie są potrzebne do przeprowadzenia rozgłaszania. Druga, podobnie z resztą jak pierwsza, jest rzadko spotykana.

```

Dane niejawne:  $G(V, E)$ 
Dane jawne:  $s, m = b_1 b_2 \dots b_r$ 

/* faza 1 */
s nadaje  $m$ ;

for  $i = 1, 2, \dots, r$  do /* kolejne fazy */

    foreach  $v$ , który poznał  $m$  w fazie  $i - 1$  do /* przekazanie
        zakodowanego  $m$  w ciągu  $2r + 4$  rund */

        /* 2 rundy na oznaczenie początku transmisji */
         $v$  nadaje  $\mu$ ;
         $v$  nadaje  $\mu$ ;

        for  $j = 1, 2, \dots, r$  do /*  $2r$  rund na zakodowanie  $r$  bitów  $m$  */

            switch  $b_j$  do
                case  $0$ 
                     $v$  nie nadaje;
                     $v$  nie nadaje;
                case  $1$ 
                     $v$  nie nadaje;
                     $v$  nadaje  $\mu$ ;

            /* 2 rundy na oznaczenie końca transmisji */
             $v$  nadaje  $\mu$ ;
             $v$  nadaje  $\mu$ ;

```

Protokół 2: Protokół ENCODED-BROADCAST

Twierdzenie 2. Protokół ENCODED-BROADCAST przeprowadza rozgłaszanie i terminuje w czasie $\mathcal{O}(|m|D)$ dla dowolnego grafu o n wierzchołkach. Co więcej, twierdzenie pozostaje w mocy nawet jeśli węzły sieci nie posiadają identyfikatorów.

3. PROSTY PROTOKÓŁ I JEGO ROZWINIĘCIA

3.1. Protokół ROUND-ROBIN. Przedstawiamy bardzo prosty protokół, o nazwie ROUND-ROBIN. Składa się on z pewnej liczby identycznych faz. Różna jest tylko faza pierwsza: źródło nadaje w niej m . Każda kolejna faza składa się z n rund. W rundzie i węzeł i nadaje wiadomość m , o ile tylko wcześniej ją otrzymał. W oczywisty sposób unikamy kolizji. Po zakończeniu fazy k , m znają wszystkie węzły w odległości co najwyżej k od źródła. Zatem do powiadomienia wszystkich węzłów wystarczy D faz, o ile znane jest n . Niestety, gdy n nie jest znane, nie wiemy ile faz potrzeba i dla hipotetycznej liczby węzłów n musimy wykonać $n - 1$ faz. Nie tylko może być $D = n - 1$, ale dodatkowo węzły na ścieżce (w tym wypadku graf jest skierowaną ścieżką długości $n - 1$) mogą być w takiej kolejności, że w jednej fazie m poznaje tylko jeden nowy węzeł. Dlatego w schemacie podwajania (i pseudokodzie) liczba faz wynosi $n - 1$, a stąd czas powiadomienia — $\mathcal{O}(n^2)$ a nie $\mathcal{O}(nD)$.

Zauważmy jeszcze, że każdy węzeł po tym, gdy nadał wiadomość, może się zdezaktywować, gdyż wie, że został usłyszany. Dotyczy to zarówno wywołania dla

określonego n , jak i schematu podwajania. Modyfikacja ta zapewnia dodatkowo, że protokół terminuje, a jego czas terminacji wynosi, tak samo jak czas powiadomienia, $\mathcal{O}(n^2)$. Pseudokod prezentujemy dla tak zmodyfikowanej wersji protokołu.

Można zezwolić węzłowi v , który otrzymał (po raz pierwszy) m w fazie k w rundzie wcześniejszej niż k , by nadał m i zdezaktywował się jeszcze w rundzie k . Jest to usprawnienie, które może przyspieszyć protokół, lecz nie musi — ma dokładnie tę samą gwarancję na czas powiadomienia i terminacji. W pseudokodzie nie uwzględniamy tej heurystyki, by zwiększyć czytelność.

<p>Dane niejawne: $G(V, E)$ Dane jawne: n, s, m</p> <pre> /* faza 1 */ s nadaje m; for k = 2, ..., n - 1 do /* kolejne fazy */ for v = 1, 2, ..., n do /* kolejne rundy fazy k */ if węzeł v otrzymał m po raz pierwszy w fazie i - 1 then v nadaje m; </pre>

Protokół 3: Protokół ROUND-ROBIN

Twierdzenie 3. *Protokół ROUND-ROBIN przeprowadza rozgłaszanie i terminuje w czasie $\mathcal{O}(n^2)$ dla dowolnego grafu o n wierzchołkach.*

3.2. Pewien szczególny przypadek. Niełatwo wymyśleć jak poprawić Protokół ROUND-ROBIN lub podejść inaczej do problemu rozgłaszania. Rozważymy szczególny przypadek, który naprowadzi nas na właściwy trop. W Protokole ROUND-ROBIN każda faza, dzięki odpowiednio dobranym rundom, gwarantuje, że wiadomość pozna co najmniej jeden nowy węzeł. Ponieważ nie ma lepszej gwarancji na liczbę tych węzłów, wykonujemy $n - 1$ faz. Nie widać, jak poprawić gwarancję i w konsekwencji zmniejszyć liczbę faz. Może uda się za to zmniejszyć liczbę rund w fazie?

Przypuśćmy, że węzły sieci mają stopień wejściowy ograniczony przez d — intuicyjnie niski stopień wejściowy to mniej potencjalnych kolizji. O każdej z faz ROUND-ROBIN można myśleć, że gwarantuje węzłowi v , który poznał m w fazie k , że przekaże m dalej w fazie $k + 1$, gdyż nadaje bez kolizji. Spójrzmy z drugiej strony — rozważmy węzeł v , który nie zna m , ale taki, że m w poprzedniej fazie poznał co najmniej jeden jego poprzednik. Węzeł v nazwiemy *niepowiadomionym węzłem granicznym* a jego znającego m poprzednika *powiadomionym węzłem granicznym*. Gwarantujemy, że w fazie $k + 1$ co najmniej jeden (a nawet że każdy) ze znających m poprzedników v , nada m bez kolizji.

Widać już, że gdy wiemy, że $\forall v \in V \deg_{in}(v) \leq d$, wspomniany warunek może być łatwiejszy do spełnienia — a w konsekwencji można zmniejszyć liczbę rund w fazie. Poniżej prezentujemy przykładowe rozwiązanie dla $d = 1$ i $d = 2$ oraz Uogólniony Protokół ROUND-ROBIN, w którym kolejne rundy każdej z faz wyznaczone są przez pewną rodzinę podzbiorów zbioru $[n]$. Znalezienie rodzin o małej mocy i użytecznych własnościach oraz ich umiejętne wykorzystanie będzie naszym dalszym celem.

W trywialnym przypadku $d = 1$ wystarczy jedna runda, w której nadają wszystkie węzły znające m , gdyż nie mogą się zagłuszyć. Dla $d = 2$ jest nieco trudniej. Załóżmy dla uproszczenia, że $n = 2^m - 1$. Wtedy wystarczy $2m = d \log(n + 1)$

```

Dane niejawne:  $G(V, E)$ 
Dane jawne:  $n, s, m$ 
Dane pomocnicze:  $\mathcal{F} = \{F_1, F_2, \dots, F_{|\mathcal{F}|}\}$ 

/* faza 1 */
s nadaje  $m$ ;
for  $k = 2, 3, \dots, n - 1$  do /* kolejne fazy */
    for  $i = 1, 2, \dots, |\mathcal{F}|$  do /* kolejne rundy fazy  $k$  */
        foreach  $v \in F_i$  do
            if węzeł  $v$  otrzymał  $m$  po raz pierwszy w fazie  $k - 1$  then
                 $v$  nadaje  $m$ ;

```

Protokół 4: Uogólniony Protokół ROUND-ROBIN

rund — w rundzie $2i - 1$ nadają węzły znające m , których i -ty bit identyfikatora wynosi 0, a w rundzie $2i$ — te które znają m a i -ty bit ich identyfikatora wynosi 1. Rozważmy niepowiadomiony v graniczny, który ma dwóch przodków znających m (gdy ma tylko jednego, analiza jest trywialna): u oraz u' . Ponieważ $u \neq u'$, istnieje bit, który je różni. Dla ustalenia uwagi niech to będzie bit i -ty. W takim razie jeden spośród u, u' przekazuje m do v w rundzie $2i - 1$, a drugi w rundzie $2i$.

Jak widać, maksymalny stopień wejściowy ma wpływ na liczbę potrzebnych rund w fazie. Ma również wpływ na to, które węzły powinny jednocześnie nadawać: w Protokole ROUND-ROBIN w każdej rundzie nadaje tylko jeden węzeł, w przykładzie dla $d = 1$ — wszystkie znające m , zaś dla $d = 2$ tylko niektóre z nich. Wystarczy, by w każdej rundzie nadawał taki podzbiór węzłów znających m , by rodzina wszystkich tych podzbiorów dla kolejnych faz gwarantowała, że pewien niepowiadomiony węzeł graniczny pozna m . W naturalny sposób otrzymujemy definicję *rodziny selektywnej*.

Nim podamy definicję, zaznaczmy, że również w Uogólnionym Protokole ROUND-ROBIN węzeł może niekiedy wykonać swoją fazę transmisji i dezaktywować się w tej samej fazie, w której (po raz pierwszy) otrzymał m . Ponownie wystarczy, by nie „ominął swojej kolejki”, tj. jeśli otrzymał m w rundzie i , by nie należał do żadnego ze zbiorów F_j dla $j \leq i$. Dla czytelności nie uwzględniamy tej heurystyki w pseudokodzie.

3.3. Rodziny selektywne, silnie selektywne oraz selektory.

Definicja 1. Rodziną (n, k) -selektywną nazywamy rodzinę \mathcal{F} podzbiorów zbioru $[n]$ taką, że

$$\forall X \subseteq [n], |X| \leq k \exists S \in \mathcal{F} |X \cap S| = 1.$$

O zbiorze S powiemy, że wybiera X .

Położmy $\mathcal{U} = V$, $k = d$ oraz niech zbiory należące do rodziny \mathcal{F} będą zbiorami węzłów nadających w kolejnych rundach każdej fazy (pod warunkiem, że znają m). Definicja ta gwarantuje żadaną własność rodzin dla Uogólnionego Protokołu ROUND-ROBIN: aż do ukończenia rozgłaszania w każdej fazie pewien niepowiadomiony węzeł graniczny pozna m .

W rzeczywistości m poznają wszystkie niepowiadomione węzły graniczne, ale nie wiemy ile ich jest. Ponieważ moc rodziny (n, k) -selektywnej \mathcal{F} jest liczbą rund w fazie dla grafów o maksymalnym stopniu wejściowym nie większym niż k , chcemy uzyskać rodziny (n, k) -selektywne o minimalnej mocy. Gdy nie zakładamy nic o stopniu wejściowym węzłów, interesują nas rodziny $(n, n - 1)$ -selektywne. Choć

pokażemy, że dla $k = d = \Theta(n)$ nie można uzyskać $|\mathcal{F}| = o(n)$, sprytniejsze zastosowanie rodzin selektywnych i tak pozwoli uzyskać lepsze rezultaty.

Z powyższych powodów godne zainteresowania są zarówno górne, jak i dolne ograniczenia na rozmiar rodzin selektywnych. Pokażemy później, jak z dolnych ograniczeń na ich rozmiar wywieść dolne ograniczenia na czas powiadomienia protokołów rozgłaszania. Tymczasem wprowadźmy jeszcze dwie struktury o zbliżonych własnościach.

Definicja 2. *Rodziną silnie (n, k) -selektywną nazywamy rodzinę \mathcal{F} podzbiorów zbioru $[n]$ taką, że*

$$\forall X \subseteq [n], |X| \leq k \forall x \in X \exists S \in \mathcal{F} X \cap S = \{x\} ,$$

tj. rodzina silnie selektywna różni się od rodziny selektywnej tym, że można wskazać konkretny element zbioru X , który będzie jedynym elementem przekroju X z pewnym zbiorem z rodziny \mathcal{F} .

Definicja 3. *(n, k) -selektorem nazywamy rodzinę \mathcal{F} podzbiorów zbioru $[n]$ taką, że*

$$\forall X, Y \subseteq [n], X \cap Y = \emptyset, \frac{k}{2} < |X| \leq k, |Y| \leq k \exists S \in \mathcal{F} |X \cap S| = 1 \wedge Y \cap S = \emptyset .$$

O zbiorze S powiemy, że wybiera X i omija Y .

Selektor jest konstrukcją podobną do dwóch poprzednich, choć jego własności są nieco inne. Po pierwsze, moc zbiorów X ograniczamy także z dołu, po drugie, co ważniejsze, istnieje zbiór S , który ma dokładnie jeden wspólny element z X , ale też nie ma wspólnych elementów z pewnym zbiorem Y . Poniżej prezentujemy dwa fakty, wynikające wprost z powyższych definicji.

Fakt 1. *Każda rodzina silnie (n, k) -selektywna \mathcal{F} , jest rodziną (n, k) -selektywną.*

Fakt 2. *Niech \mathcal{S}_i , dla $i = 0, 1, \dots, \lceil \log k \rceil$ będzie $(n, 2^i)$ -selektorem. Wtedy $\mathcal{F} = \bigcup_{i=0}^{\lceil \log k \rceil} \mathcal{S}_i$ jest rodziną (n, k) -selektywną.*

Oba fakty wykorzystują tylko część własności rodzin silnie selektywnych i selektorów. Z powodu słabszych własności rodzin selektywnych, nie widać, jak dzięki nim uzyskać rodziny silnie selektywne lub selektory.

Rodziny selektywne zostały wprowadzone w [CGGPR00], jednej z pierwszych prac poświęconych rozgłaszaniu w sieciach radiowych. Dzięki rodzinom selektywnym uzyskano tam pierwszy protokół o czasie powiadomienia $o(n^2)$, dokładnie $\mathcal{O}\left(n^{\frac{11}{6}}\right)$. Selektory zostały wprowadzone w [CGR00] i pozwoliły uzyskać jeden z najlepszych znanych obecnie wyników. Przedstawiamy go w dalszej części pracy, ale już teraz zaznaczamy, że czas powiadomienia protokołu z [CGR00] zależy od rozmiaru selektorów.

Nasza definicja selektora różni się nieznacznie od tej z [CGR00] i [Ind02], mianowicie u nas moc zbioru X spełnia nierówność $\frac{k}{2} < |X| \leq k$, zaś we wspomnianych pracach było to $\frac{k}{2} \leq |X| \leq k$.

Czasami rodziny selektywne i selektory będziemy zapisywać nie jako rodziny podzbiorów $[n]$ a jako ciągi tych podzbiorów. Bierze się to z tego, że będziemy potrzebować, by były one w określonym porządku, tak jak w Uogólnionym Protokole ROUND-ROBIN.

3.4. Niekonstruktywne ograniczenia górne na rozmiary rodzin selektywnych i selektorów. Pokażemy teraz, że istnieją rodziny (n, d) -selektywne, oraz (n, d) -selektory rozmiaru $\mathcal{O}(d \log n)$. Mimo, że z faktu istnienia (n, d) -selektorów rozmiaru $\mathcal{O}(d \log n)$, na podstawie faktu 2 wynika istnienie rodzin (n, d) -selektywnych tego samego rozmiaru, podamy oba dowody. Oba są przykładem zastosowania metody probabilistycznej, nie dają więc konstrukcji tych obiektów. Twierdzenie

dotyczące rodzin selektywnych pochodzi z [CMS01], zaś dotyczące selektorów z [CGR00]. Na koniec podamy odrobinę silniejsze ograniczenie górne dla rodzin selektywnych, pochodzące z [CMS03].

Twierdzenie 4. *Dla każdych $n, k \in \mathbb{N}$, $2 \leq k \leq n$ istnieje rodzina (n, d) -selektywna \mathcal{F} taka, że $|\mathcal{F}| = \mathcal{O}(k \log n)$.*

Dowód. Wylosujmy elementy rodziny \mathcal{F} . Przypominamy, że każdy z nich jest podzbiorem $[n]$. Wylosujemy m takich zbiorów, gdzie wartość m ustalimy później, w następujący sposób: $\mathcal{F} = \{F_i\}_{i=1}^m$. Dla $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, n$, $\Pr[j \in F_i] = 1/k$, gdzie losowania są niezależne. Oszacujemy teraz prawdopodobieństwo, że \mathcal{F} nie jest rodziną (n, k) -selektywną, czyli że istnieje zbiór $W \subseteq [n]$, $|W| = h \leq k$ taki, że żaden zbiór $F \in \mathcal{F}$ nie wybiera W .

$$\begin{aligned} \Pr[\text{BAD}] &= \Pr[\mathcal{F} \text{ nie jest rodziną } (n, k)\text{-selektywną}] \leq \\ &\leq \sum_{h=1}^k \binom{n}{h} \left(1 - \frac{h}{k} \left(1 - \frac{1}{k}\right)^{h-1}\right)^m \leq \sum_{h=1}^k n^h \left(1 - \frac{h}{k} \left(1 - \frac{1}{k}\right)^k\right)^m \end{aligned}$$

Ponieważ $(1 - \frac{1}{k})^k \geq \frac{1}{4}$ (jest to ciąg rosnący) i $1 + t \leq e^t$, mamy

$$\Pr[\text{BAD}] \leq \sum_{h=1}^k n^h \left(1 - \frac{h}{4k}\right)^m \leq \sum_{h=1}^k n^h e^{-\frac{h}{4k}m}.$$

Położmy teraz $m = 4k\beta \log n$, gdzie $\beta > 2$. Wtedy

$$\Pr[\text{BAD}] \leq \sum_{h=1}^k n^h e^{-\beta h \log n} = \sum_{h=1}^k n^{h-h\beta} \leq \sum_{h=1}^k \frac{1}{n^{\beta-1}} = \frac{1}{n^{\beta-2}} = \frac{1}{n^{\Theta(1)}}.$$

Utworzona w ten sposób rodzina \mathcal{F} spełnia $|\mathcal{F}| = \mathcal{O}(k \log n)$ oraz z dużym prawdopodobieństwem jest rodziną (n, k) -selektywną. \square

Twierdzenie 5. *Dla każdych $n, k \in \mathbb{N}$, $2 \leq k \leq n$ istnieje (n, k) -selektor \mathcal{S} taki, że $|\mathcal{S}| = \mathcal{O}(k \log n)$.*

Dowód. Losujemy m elementów selektora \mathcal{S} podobnie jak poprzednio, ale z nieco innym rozkładem prawdopodobieństwa: $\mathcal{S} = \{S_i\}_{i=1}^m$. Dla $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, n$, $\Pr[j \in S_i] = 1/k + 1$, gdzie losowania są niezależne. Ustalmy $i \in [m]$ oraz rozłączne zbiory X i Y takie, że ich moce $x = |X|$ oraz $y = |Y|$ spełniają nierówności: $k/2 \leq x \leq k$ oraz $y \leq k$. Wtedy

$$\begin{aligned} \Pr[|S_i \cap X| = 1 \wedge S_i \cap Y = \emptyset] &= \\ &= x \cdot \frac{1}{k+1} \cdot \left(1 - \frac{1}{k+1}\right)^{x-1} \left(1 - \frac{1}{k+1}\right)^y = \\ &= \frac{x}{k} \left(1 - \frac{1}{k+1}\right)^{x+y} \geq \frac{1}{2} \left(1 - \frac{1}{k+1}\right)^{2(k+1)} \geq \frac{1}{32}. \end{aligned}$$

Ostatnia nierówność, jak poprzednio, wynika z monotoniczności ciągu $(1 - 1/k)^k$. Z niezależności zbiorów S_i dostajemy

$$\begin{aligned} \Pr[\mathcal{S} \text{ nie jest } (n, k)\text{-selektorem}] &\leq \\ &\leq \sum_{x=k/2}^k \binom{n}{x} \sum_{y=0}^k \binom{n}{y} \left(\frac{31}{32}\right)^m \leq k^2 n^{2k} \left(\frac{31}{32}\right)^m \leq n^{4k} \left(\frac{31}{32}\right)^m \leq n^{-\beta} = \frac{1}{n^{\Theta(1)}}, \end{aligned}$$

jeśli położymy $m = \frac{1}{\log \frac{32}{31}} (4k + \beta) \log n$. \square

W późniejszej pracy [CMS03] pokazano lepsze ograniczenie górne na rozmiar rodziny selektywnej. Wynik ten prezentujemy zaraz po dwóch prostych faktach, z których korzysta. Fakty te uzupełniamy o dowody, pominięte w [CMS03].

Fakt 3. Dla $n, k \in \mathbb{N}$, $k < n$, zachodzi $\log \binom{n}{k} = \Theta(k \log \frac{n}{k} + k)$.

Dowód. Logarytmując nierówność

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n \cdot e}{k}\right)^k,$$

otrzymujemy

$$k \log \frac{n}{k} \leq \log \binom{n}{k} \leq k \log \frac{n}{k} + k \log e.$$

□

Fakt 4. Dla $n, h \in \mathbb{N}$, takich, że $2^h \leq n$, zachodzi $\sum_{i=0}^h 2^i \log \frac{n}{2^i} = \Theta(2^h \log \frac{n}{2^h} + 2^h)$.

Dowód.

$$(1) \quad \sum_{i=0}^h 2^i \log \frac{n}{2^i} = \log n \sum_{i=0}^h 2^i - \sum_{i=0}^h i 2^i = (2^{h+1} - 1) \log n - \sum_{i=0}^h i 2^i$$

By otrzymać wartość $\sum_{i=0}^h i 2^i$, rozważmy sumę $\sum_{i=0}^h i x^i$:

$$\begin{aligned} \sum_{i=0}^h i x^i &= \sum_{i=1}^h i x^i = \sum_{i=1}^h x \cdot (x^i)' = x \left(\sum_{i=1}^h x^i \right)' = x \left(\sum_{i=0}^h x^i \right)' = x \left(\frac{x^{h+1} - 1}{x - 1} \right)' = \\ &= x \cdot \frac{(h+1)x^h(x-1) - (x^{h+1} - 1)}{(x-1)^2} = x \cdot \frac{h \cdot x^{h+1} - (h+1)x^h + 1}{(x-1)^2}. \end{aligned}$$

Podstawiając $x = 2$ otrzymujemy

$$\sum_{i=0}^h i 2^i = 2(h2^{h+1} - (h+1)2^h + 1) = 2^{h+1}(2h - (h+1)) + 2 = 2^{h+1}(h-1) + 2.$$

Wstawiając powyższy wynik do (1), uzyskujemy

$$\sum_{i=0}^h 2^i \log \frac{n}{2^i} = (2^{h+1} - 1) \log n - 2^{h+1}(h-1) - 2 = 2^{h+1}(\log n - h + 1) + 2 - \log n.$$

□

Twierdzenie 6. Dla każdych $n, k \in \mathbb{N}$, $2 \leq k < n$ istnieje rodzina (n, k) -selektywna \mathcal{F} taka, że $|\mathcal{F}| = \mathcal{O}(k \log \frac{n}{k} + k)$.

Dowód. Zmodyfikujmy lekko Definicję 1 — powiemy, że rodzina \mathcal{F} jest selektywna dla rodziny \mathcal{S} , jeśli dla każdego $S \in \mathcal{S}$ istnieje $F \in \mathcal{F}$ taki, że $|S \cap F| = 1$.

Niech \mathcal{S}_i dla $i = 1, 2, \dots, \lceil \log k \rceil$ będzie rodziną wszystkich podzbiorów $[n]$ o mocy z przedziału $(2^{i-1}, 2^i]$. Rozważmy rodzinę \mathcal{F}_i o mocy l_i (które ustalimy później), której każdy zbiór F losujemy następująco: dla każdego $x \in [n]$ niezależnie określamy, że $x \in F$ z prawdopodobieństwem $\frac{1}{2^i}$. Mamy więc $\forall F \in \mathcal{F}_i \forall x \in [n] \Pr[x \in F] = \frac{1}{2^i}$.

Ustalmy zbiór $S \in \mathcal{S}_i$ i rozważmy zbiór $F \in \mathcal{F}_i$. Wtedy

$$\Pr[|S \cap F| = 1] = \frac{|S|}{2^i} \left(1 - \frac{1}{2^i}\right)^{|S|-1} > \frac{|S|}{2^i} \left(1 - \frac{1}{2^i}\right)^{2^i} \geq \frac{|S|}{4 \cdot 2^i} \geq \frac{1}{8},$$

gdzie środkowa nierówność ponownie wynika stąd, że ciąg $(1 - \frac{1}{k})^k$ jest rosnący. Ponieważ zbiory z \mathcal{F}_i losujemy niezależnie, prawdopodobieństwo, że \mathcal{F}_i nie jest rodziną selektywną dla \mathcal{S}_i , szacujemy przez

$$\begin{aligned} \Pr[\text{BAD}] &= \Pr[\mathcal{F}_i \text{ nie jest rodziną selektywną dla } \mathcal{S}_i] \leq \\ &\leq \sum_{S \in \mathcal{S}_i} \Pr[\mathcal{F}_i \text{ nie wybiera } S] \leq \sum_{d=2^{i-1}+1}^{2^i} \binom{n}{d} \left(1 - \frac{1}{8}\right)^{l_i} \leq \sum_{d=2^{i-1}+1}^{2^i} \binom{n}{d} e^{-\frac{l_i}{8}}. \end{aligned}$$

Wystarczy wybrać $l_i = 8 \ln \left(\binom{n}{2^i} 2^i\right)$, by dostać

$$\Pr[\text{BAD}] \leq \sum_{d=2^{i-1}+1}^{2^i} \binom{n}{d} e^{-\frac{l_i}{8}} = \sum_{d=2^{i-1}+1}^{2^i} \frac{\binom{n}{d}}{\binom{n}{2^i} 2^i} \leq \sum_{d=2^{i-1}+1}^{2^i} \frac{1}{2^i} = \frac{2^{i-1}}{2^i} = \frac{1}{2},$$

gdzie nierówność pomiędzy skrącami symbolami Newtona wynika z nierówności $d \leq 2^i \leq \frac{n}{2}$.

Powyższa nierówność pokazuje, że z niezerowym prawdopodobieństwem wylosujemy \mathcal{F}_i będące rodziną selektywną dla \mathcal{S}_i . Teraz pozostaje zająć się jej rozmiarem. Z Faktu 3 wynika, że $\log \binom{n}{t} = \mathcal{O}(t \log \frac{n}{t} + t)$, a stąd bezpośrednio, że $l_i \leq c 2^i (\log \frac{n}{2^i} + 1)$ dla pewnej stałej $c > 0$, czyli $|\mathcal{F}_i| \leq c 2^i (\log \frac{n}{2^i} + 1)$ dla tej samej stałej c . Wreszcie, rodzina

$$\mathcal{F} = \bigcup_{i=1}^{\lceil \log k \rceil} \mathcal{F}_i$$

jest rodziną (n, k) -selektywną, zaś jej rozmiar, na mocy Faktu 4, wynosi

$$|\mathcal{F}| \leq \sum_{i=1}^{\lceil \log k \rceil} |\mathcal{F}_i| \leq \sum_{i=1}^{\lceil \log k \rceil} c 2^i \left(\log \frac{n}{2^i} + 1\right) = \mathcal{O}\left(k \log \frac{n}{k} + k\right).$$

□

Warto zauważyć, że w konstrukcji rodziny selektywnej w powyższym dowodzie osobno rozważaliśmy zbiory X o mocach z przedziałów $(2^i, 2^{i+1}]$, tj. faktycznie zajmowaliśmy się zbiorami X jak w definicji selektora (Definicja 3). Ostatecznie rodzinę selektywną otrzymaliśmy jako sumę podobną do tej z Faktu 2. To pomysł z selektorów, by osobno traktować zbiory X o mocach z przedziałów $(2^i, 2^{i+1}]$, pozwolił nam poprawić ograniczenie.

W [CMS03] w powyższych faktach i twierdzeniu w ograniczeniach nie występował składnik k (ew. 2^h). Dodaliśmy go, by uniknąć niejasności dla k bliskiego n — wtedy $\log \frac{n}{k}$ może być dowolnie bliskie 0. Oczywiście najważniejsze jest ograniczenie górne na rozmiar rodziny (n, k) -selektywnej. Dla dowolnego k taką rodziną jest rodzina wszystkich n singletonów z $[n]$, więc $k = \Theta(n)$ nie jest najważniejszym dla nas przypadkiem.

Potencjalnie jest możliwe, że istnieją rodziny selektywne lub selektory o jeszcze lepszych parametrach niż uzyskane. Z tego powodu interesują nas ograniczenia dolne na ich rozmiary. Podajemy je w dalszej części pracy. Okazuje się, że pokazane powyżej ograniczenia górne na rozmiar rodzin selektywnych są optymalne, zaś selektorów prawie optymalne.

Pokazaliśmy, że istnieją rodziny selektywne i selektory niewielkich rozmiarów, a nawet że można je łatwo wylosować z dużym prawdopodobieństwem sukcesu. Daje nam to wyłącznie ograniczenie górne na rozmiary tych obiektów, bez możliwości zastosowania ich w deterministycznych protokołach rozgłaszania. Z drugiej strony, nawet gdyby dopuścić możliwość losowania rodzin selektywnych (selektorów), trudno byłoby zapewnić, że każdy węzeł wylosuje tę samą rodzinę (selektor).

Dlatego wciąż interesują nas konstrukcje *explicite*, o tych samych, lub niewiele większych rozmiarach.

Rodziny silnie selektywne wspominamy z dwóch powodów: znajdują podobne zastosowanie jak rodziny selektywne w ogólniejszym problemie *multibroadcast* a dodatkowo ich konstrukcje znane są od lat. W naturalny sposób dostarczały również konstrukcji rodzin selektywnych, jednak o zbyt dużych rozmiarach. Celowo pomijamy górne ograniczenia na rozmiar rodzin silnie selektywnych, bowiem dolne ograniczenia na rozmiar rodzin silnie selektywnych są dużo większe niż rozmiary znanych obecnie konstrukcji selektorów. Więcej na ten temat w dalszej części pracy.

4. NIETRYWIALNE GÓRNE OGRANICZENIA NA CZAS ROZGLĄSZANIA

W tym rozdziale zaprezentujemy dwa wyniki. Pierwszy z nich to protokół z [CMS01], który gwarantuje czas powiadomienia uzależniony od trzech parametrów grafu: n , D , d . Jego czas powiadomienia wynosi $\mathcal{O}(Dd \log \frac{n}{d} \log^3 n)$. Jakkolwiek daje to czas $\omega(n^2)$ gdy $D = \Omega(n)$, $d = \Omega(n)$, to nie zawiera n jako czynnika liniowego, przez co daje przyzwoite wyniki gdy d i D są małe.

Drugi protokół pochodzi z [CGR00] i jest jednym z najszybszych protokołów — jego czas powiadomienia wynosi $\mathcal{O}(n \log^2 n)$. Jak się później okaże, jest to bardzo dobry wynik, niestety zależy tylko od n , tj. dla wspomnianych małych D i d może nie być konkurencyjny. Wynik ten został poprawiony później w [KP03b] do silniej zależącego od parametrów grafu $\mathcal{O}(n \log n \log D)$.

4.1. Protokół zależny od n , D i d . W poprzednim rozdziale motywacją dla wprowadzenia i badania rodzin selektywnych była próba przyspieszenia Protokołu ROUND-ROBIN dla grafów o ograniczonym stopniu wejściowym wierzchołków. Teraz rozwinie ten pomysł w oparciu o [CMS01], poprawiając drobne błędy¹, które tam występowały.

Chcemy opracować kolejne protokoły BROAD-A, BROAD-B i BROAD-C, które przeprowadzą rozgłaszanie dysponując coraz mniejszą wiedzą na temat grafu. BROAD-A zakłada znajomość n i d , BROAD-B tylko n , zaś BROAD-C nie zakłada o grafie nic. Chcemy przy tym, by czas powiadomienia tych protokołów istotnie zależał od tych trzech parametrów, w szczególności by n nie występowało w nim jako czynnik liniowy.

4.1.1. Protokół zależny od n i d . Protokół BROAD-A właściwie opracowaliśmy już wcześniej — to Uogólniony Protokół ROUND-ROBIN, którego fazy wyznacza rodzina (n, d) -selektywna. Jak poprzednio, każdy węzeł po wykonaniu pełnej fazy transmisji dezaktywuje się. Przypominamy, że protokół składa się z kolejnych faz, z których każda składa się z tylu rund, ile wynosi moc rodziny (n, d) -selektywnej $\mathcal{F} = \{F_1, F_2, \dots, F_{|\mathcal{F}|}\}$. Z twierdzenia 6 wynika, że możemy założyć, że $|\mathcal{F}| = \mathcal{O}(d \log \frac{n}{d} + d)$. W rundzie j fazy i nadają te węzły, które należą do F_j oraz odebrały m (po raz pierwszy) w fazie $i - 1$. Jak widać, każdy węzeł nadaje powyżej jednej fazy. Dla porządku ponownie zamieszczamy pseudokod protokołu. Z wcześniejszej dyskusji o Uogólnionym Protokole ROUND-ROBIN i rodzinach selektywnych, otrzymujemy:

Twierdzenie 7. *Protokół BROAD-A uruchomiony na dowolnym grafie o n wierzchołkach, maksymalnym stopniu wejściowym d i maksymalnej odległości od źródła D przeprowadza rozgłaszanie i terminuje w czasie $\mathcal{O}(Dd(\log \frac{n}{d} + 1))$.*

¹O błędach i ich korekcie więcej na końcu podrozdziału.

Dane niejawne: $G(V, E)$ Dane jawne: n, d, s, m Dane pomocnicze: rodzina (n, d) -selektywna $\mathcal{F} = \{F_1, F_2, \dots, F_{ \mathcal{F} }\}$
/* faza 1 */ s nadaje m ; for $i = 2, 3, \dots, n$ do /* kolejne fazy */ for $j = 1, 2, \dots, \mathcal{F} $ do /* kolejne rundy fazy i */ foreach $v \in F_j$ do if węzeł v otrzymał m po raz pierwszy w fazie $i - 1$ then v nadaje m

Protokół 5: Protokół BROAD-A

4.1.2. *Protokół zależny od n .* Na bazie BROAD-A opracujemy BROAD-B. Ustalmy n , bo z tej wielkości BROAD-B może korzystać. Zauważmy, że mamy podobną sytuację co w ROUND-ROBIN — jeśli zastosować schemat podwajania ze względu na d dla BROAD-A, dla każdej wartości d musimy wykonać n faz, bo nie znamy D . Z Faktu 4 czas powiadomienia wynosiłby $\mathcal{O}(nd(\log \frac{n}{d} + 1))$. Chcieliśmy uniknąć liniowej zależności od n , więc musimy szukać innego rozwiązania.

Zamiast wykonywać protokoły BROAD-A($n, 2^i$) dla $i = 0, 1, \dots, \lceil \log n \rceil$ sekwencyjnie, uruchomimy je równoległe, przeplatając rundy każdego z nich. Zakładamy, że równoległe wywołania protokołu BROAD-A dla różnych d są częściowo zależne — jeśli węzeł v pozna m w rundzie t protokołu BROAD-A($n, 2^i$), to pozna m również w rundzie t w każdym z równoległych wywołań BROAD-A($n, 2^j$) dla wszystkich j . Służy to jedynie redukcji czasu terminacji. Podobnie będzie w protokole BROAD-C: oprzemy go przecież na protokole BROAD-B — tam również węzły „nie będą kryć swojej wiedzy”.

Ostatecznie więc Protokół BROAD-B składa się z kolejnych faz numerowanych od 1. Każda faza składa się z $\lceil \log n \rceil$ rund, numerowanych od 1. W rundzie j fazy i węzły wykonują rundę i Protokołu BROAD-A dla $d = 2^j$. Oczywiście można pierwszą fazę uczynić wspólną dla wszystkich równoległych wywołań BROAD-A, bo w jej jedynej rundzie s nadaje m . Nie czynimy tego w pseudokodzie, by skrócić zapis.

Dane niejawne: $G(V, E)$ Dane jawne: n, s, m
for $i = 1, 2, \dots$ do /* kolejne fazy */ for $j = 1, 2, \dots, \lceil \log n \rceil$ do /* kolejne rundy fazy i ; w rundzie j zakładany stopień wejściowy nie większy niż 2^j */ wykonaj i -tą rundę protokołu BROAD-A($n, 2^j$);

Protokół 6: Protokół BROAD-B

Twierdzenie 8. *Protokół BROAD-B uruchomiony na dowolnym grafie o n wierzchołkach, maksymalnym stopniu wejściowym d i maksymalnej odległości od źródła D przeprowadza rozgłaszanie w czasie $\mathcal{O}(Dd \log n (\log \frac{n}{d} + 1))$ i terminuje w czasie $\mathcal{O}(Dd \log n (\log \frac{n}{d} + 1) + n \log n)$.*

Dowód. Na mocy Twierdzenia 7 wykonanie BROAD-A($n, 2^{\lceil \log d \rceil}$) sprawia, że wszystkie węzły poznają m . W Protokole BROAD-B zajmie to $\lceil \log n \rceil$ razy więcej czasu, bo symulujemy równoległe wykonanie $\lceil \log n \rceil$ protokołów. Stąd czas powiadomienia BROAD-B wynosi $\mathcal{O}(Dd \log n (\log \frac{n}{d} + 1))$. Również czas dezaktywacji węzła po otrzymaniu m wydłuża się $\lceil \log n \rceil$ razy. W protokole BROAD-A($n, 2^j$) wynosi on $\mathcal{O}(2^j \log \frac{n}{2^j} + 2^j)$, co dla maksymalnej wartości $j = \lceil \log n \rceil$ daje $\mathcal{O}(n)$. Stąd czas terminacji BROAD-B wynosi $\mathcal{O}(Dd \log n (\log \frac{n}{d} + 1) + n \log n)$. \square

4.1.3. Ostateczny protokół. BROAD-C można łatwo uzyskać wykonując kolejne fazy takie, że faza i składa się z i rund a runda j należy do protokołu BROAD-B(2^j). Niestety, równie łatwo stwierdzić, że wtedy czas powiadomienia i czas terminacji BROAD-C to odpowiednio te czasy dla BROAD-B podniesione do kwadratu! Pokażemy wydajniejszy sposób przeplatania protokołów BROAD-B dla różnych n .

W tym celu zdefiniujemy rodzinę funkcji

$$f_0(z) = 0, \quad f_k(z) = 2^k(k - z) \quad \text{dla } k = 1, 2, 3, \dots$$

Protokół BROAD-C składa się z k kolejnych faz, numerowanych od 1. Faza k składa się z k tur, numerowanych od 0. W turze l fazy k wykonywane są rundy $f_{k-1}(l) + 1, f_{k-1}(l) + 2, \dots, f_k(l)$ protokołu BROAD-B(2^{2^l}).

Dane niejawne: $G(V, E)$

Dane jawne: s, m

/* $f_0(z) = 0, \quad f_k(z) = 2^k(k - z) \quad \text{dla } k = 1, 2, 3, \dots$ */

for $k = 1, 2, \dots$ do /* kolejne fazy */

 for $l = 0, 1, \dots, k - 1$ do /* kolejne tury fazy k */

 for $h = f_{k-1}(l) + 1, f_{k-1}(l) + 2, \dots, f_k(l)$ do /* kolejne rundy tury l fazy k */

 wykonaj rundę h protokołu BROAD-B(2^{2^l});

Protokół 7: Protokół BROAD-C

Nas interesują oczywiście rundy protokołu BROAD-B($2^{2^{\lceil \log \log n \rceil}}$), które wykonywane są w turach $\lceil \log \log n \rceil$ kolejnych faz. Tura $\lceil \log \log n \rceil$ po raz pierwszy pojawia się w fazie $\lceil \log \log n \rceil + 1$. Z definicji funkcji $f_k(z)$ wynika, że $f_k(k) = 0$, czyli Protokół BROAD-B dla każdej wartości n wykonujemy od jego pierwszej rundy.

Twierdzenie 9. *Protokół BROAD-C uruchomiony na dowolnym grafie o n wierzchołkach, maksymalnym stopniu wejściowym d i maksymalnej odległości od źródła D przeprowadza rozgłaszanie w czasie $\mathcal{O}(Dd (\log \frac{n}{d} + 1) \log^3 n)$ i terminuje w czasie $\mathcal{O}(Dd (\log \frac{n}{d} + 1) \log^3 n + n \log^3 n)$.*

Dowód. Przeprowadzimy analizę dla czasu powiadomienia. Dla czasu terminacji analiza jest analogiczna.

Rozgłaszanie zostanie ukończone wraz z końcem wykonania BROAD-B($2^{2^{\lceil \log \log n \rceil}}$). Musimy oszacować, kiedy zostanie wykonana ostatnia runda tego protokołu. Niech jej numerem będzie t_{end} . Z Twierdzenia 8 wynika, że $t_{end} = \mathcal{O}(Dd \log n (\log \frac{n}{d} + 1))$. Zajmujemy się gwarancją na czas powiadomienia a nie faktycznym czasem, więc założymy, że $t_{end} = \Theta(Dd \log n \log(\frac{n}{d} + 1))$. Dolne ograniczenie tej wielkości będzie ważne! Zauważmy więc, że niezależnie od wartości d i D zachodzi $t_{end} = \Omega(\log^2 n)$.

Teraz oszacujemy z góry k_{end} , numer fazy protokołu BROAD-C, w której mieści się runda t_{end} . Gdy będziemy już dysponować ograniczeniem górnym na k_{end} , łatwo oszacujemy z góry liczbę rund w k_{end} pierwszych fazach.

Oczywiście k_{end} jest najmniejszą taką liczbą, że

- (1) $k_{end} > \lceil \log \log n \rceil$, bo wcześniej BROAD-B $(2^{\lceil \log \log n \rceil})$ nie jest wykonywane
- (2) $f_{k_{end}}(\lceil \log \log n \rceil) \geq t_{end}$, bo w turze $\lceil \log \log n \rceil$ fazy k_{end} wykonywane są rundy BROAD-B $(2^{\lceil \log \log n \rceil})$ aż do $f_{k_{end}}(\lceil \log \log n \rceil)$ włącznie.

Podstawiając $k = \lceil \log t_{end} \rceil$ oraz $z = \lceil \log \log n \rceil$ w definicji $f_k(z)$ otrzymujemy

$$(2) \quad f_{\lceil \log t_{end} \rceil}(\lceil \log \log n \rceil) = 2^{\lceil \log t_{end} \rceil} (\lceil \log t_{end} \rceil - \lceil \log \log n \rceil) \geq \geq t_{end} (\lceil \log t_{end} \rceil - \lceil \log \log n \rceil) \geq t_{end},$$

ponieważ $t_{end} = \Omega(\log^2 n)$. Oznacza to, że zachodzi nierówność

$$(3) \quad k_{end} \leq \lceil \log t_{end} \rceil < \log t_{end} + 1.$$

Teraz obliczymy, ile rund potrzeba na zakończenie k_{end} -tej fazy. Z definicji faz i tur wynika, że potrzeba ich

$$\begin{aligned} T &= \sum_{k=1}^{k_{end}} \sum_{l=0}^{k-1} (f_k(l) - f_{k-1}(l)) = \sum_{l=0}^{k_{end}-1} \sum_{k=l+1}^{k_{end}} (f_k(l) - f_{k-1}(l)) = \\ &= \sum_{l=0}^{k_{end}-1} (f_{k_{end}}(l) - f_l(l)) = \sum_{l=0}^{k_{end}-1} f_{k_{end}}(l). \end{aligned}$$

Podstawiając z definicji wartość $f_{k_{end}}(l)$ otrzymujemy

$$(4) \quad T = \sum_{l=0}^{k_{end}-1} f_{k_{end}}(l) = \sum_{l=0}^{k_{end}-1} 2^{k_{end}} (k_{end} - l) < 2^{k_{end}} k_{end}^2.$$

Podstawiając w nierówności (4) ograniczenie na k_{end} z nierówności (3), otrzymujemy wreszcie

$$T < 2 \cdot t_{end} (\log t_{end} + 1)^2 = \mathcal{O}(t_{end} \log^2 t_{end}) = \mathcal{O}\left(Dd \left(\log \frac{n}{d} + 1\right) \log^3 n\right).$$

Jak widać, wystarczy przyjąć za t_{end} czas terminacji Protokołu BROAD-B $(2^{2^{\lceil \log \log n \rceil}})$, tj. $t_{end} = \Theta(Dd \log n (\log \frac{n}{d} + 1) + n \log n)$, by identycznie wykazać, że protokół BROAD-C zwiększa ten czas o czynnik $\log^2 n$. \square

W [CMS01] przedstawione były błędne protokoły BROAD-B oraz BROAD-C. Nie przeprowadzały poprawnie rozgłaszania, gdyż węzły dezaktywowały się zbyt prędko. Dodatkowo o protokole BROAD-B autorzy [CMS01] twierdzili, że zarówno jego czas powiadomienia jak i terminacji wynoszą $\mathcal{O}(Dd \log n \log \frac{n}{d})$. Podobnie miało być w przypadku protokołu BROAD-C, który dodatkowo zależał od parametru α . Oba czasy miały wynosić $\mathcal{O}(Dd \log \frac{n}{d} \log^{1+\alpha} n)$ dla dowolnej stałej $\alpha > 0$. Jednocześnie tura l odpowiadać miała protokołowi BROAD-B (2^l) a nie BROAD-B (2^{2^l}) . W oszacowaniu czasu był błąd, którego poprawa wymagała nie tylko wspomnianej zmiany dotyczącej tur, ale też spełnienia nierówności $\alpha \geq 2$. Ostatecznie więc parametr α zniknął, gdyż optymalny wynik daje jego minimalna wartość. Błędy dostrzegł autor niniejszej pracy i poprawił protokół BROAD-B. Korekty protokołu BROAD-C dokonali sami autorzy [CMS01]. Błąd w szacowaniu czasu powiadomienia dla Protokołu BROAD-C był w nierówności (2). By była prawdziwa, potrzeba, by $\lceil \log t_{end} \rceil > \lceil \log \log n \rceil$, co nie było prawdą — dla małych d i D mogło być $t_{end} = \mathcal{O}(\log^2 n)$.

4.2. Szybki protokół rozgłaszania. Przedstawimy teraz protokół DOBROADCAST z [CGR00], który uzyskuje czas powiadomienia $\mathcal{O}(n \log^2 n)$, korzystając z (n, k) -selektorów rozmiaru $\mathcal{O}(k \log n)$. Twierdzenie 5 gwarantuje istnienie selektorów o takich rozmiarach, jednak nie daje konstrukcji. Dlatego dokonamy szacunku odrobinę ogólniejszego niż w [CGR00], nie zakładając nic o rozmiarach selektorów. Protokół DOBROADCAST, z czasem powiadomienia $\mathcal{O}(n \log^2 n)$ był do niedawna najszybszym znanym protokołem rozgłaszania. Na końcu podrozdziału informujemy o lepszym wyniku z [KP03b].

Dla $j = 0, 1, \dots, \lceil \log n \rceil$, niech $\mathcal{S}^j = (S_0^j, S_1^j, \dots, S_{m_j-1}^j)$ będzie $(n, 2^j)$ -selektorem, zaś m_j jego rozmiarem. Protokół DOBROADCAST składa się z kolejnych faz, numerowanych od 0, z których każda składa się z $\lceil \log n \rceil + 1$ rund, również numerowanych od 0. Zbiór węzłów nadających (pod warunkiem, że znają m) w rundzie j fazy i to $S_{i \bmod m_j}^j$. Jak widać, Protokół DO-BROADCAST polega na równoległym wykonaniu protokołów korzystających z $(n, 2^j)$ -selektorów dla różnych j .

Dane niejawne: $G(V, E)$
Dane jawne: n, s, m
Dane pomocnicze: $(n, 2^k)$ -selektory $\mathcal{S}^k = (S_0^k, S_1^k, \dots, S_{m_k-1}^k)$, dla $0 \leq k \leq \lceil \log n \rceil$

for $i = 0, 1, \dots$ **do** /* kolejne fazy */

for $j = 0, 1, \dots, \lceil \log n \rceil$ **do** /* kolejne rundy fazy i */

foreach $v \in S_{i \bmod m_j}^j$ **do**
 if v zna m **then**
 v nadaje m ;

Protokół 8: Protokół DO-BROADCAST

Podczas analizy czasu powiadomienia będziemy rozważać cztery klasy węzłów: *uśpione*, *graniczne powiadomione*, *graniczne niepowiadomione* i *przetworzone*. *Węzeł uśpiony* to węzeł nie znający m . *Węzeł graniczny powiadomiony* to węzeł znający m , którego co najmniej jeden następnik nie zna m . *Węzeł graniczny niepowiadomiony* to uśpiony węzeł, którego co najmniej jeden poprzednik zna m . *Węzeł przetworzony* to węzeł, który zna m i którego wszyscy następnicy również je znają. Każda zmiana stanu węzła z uśpionego na brzegowy powiadomiony, lub z brzegowego powiadomionego na przetworzony, wnosi 1 do *postępu*. Zauważmy, że obie te zmiany mogą nastąpić w jednej rundzie. Gdy postęp wyniesie $2n - 1$, wszystkie węzły znają m .

Pokażemy, że dla dowolnej fazy i istnieje j takie, że zamortyzowany postęp w każdej z faz $i, i + 1, \dots, i' = i + m_j - 1$ wynosi $\Omega\left(\frac{2^j}{m_j}\right)$. Stąd dostaniemy, że czas powiadomienia wynosi $\mathcal{O}(n \log n \cdot \max_j \frac{m_j}{2^j})$. W szczególności dla (n, k) -selektorów rozmiaru $\mathcal{O}(n \log^c n)$ mamy $m_j = \mathcal{O}(2^j \log^c n)$, a więc czas powiadomienia wyniesie $\mathcal{O}(n \log^{1+c} n)$.

Lemat 1. *Dla dowolnej fazy i protokołu DO-BROADCAST istnieje j takie, że zamortyzowany postęp w każdej z faz $i, i + 1, \dots, i' = i + m_j - 1$ wynosi $\Omega\left(\frac{2^j}{m_j}\right)$.*

Dowód. Niech B będzie zbiorem powiadomionych węzłów granicznych tuż przed rozpoczęciem fazy i , zaś b niech będzie takie, że $2^{b-1} < |B| \leq 2^b$. Dla $j = 1, 2, \dots$ niech Y_j będzie zbiorem węzłów, które były uśpione przed rozpoczęciem fazy i a po

raz pierwszy otrzymały m w jednej z faz $i, i+1, \dots, i+m_j-1$. Rozważmy dwa przypadki:

- (1) Istnieje j takie, że $|Y_j| \geq 2^j$. Wtedy po m_j rundach postęp wynosi co najmniej 2^j . Zamortyzowany postęp wynosi więc co najmniej $2^j/m_j$.
- (2) Dla każdego j zachodzi $|Y_j| < 2^j$. Pokażemy, że wszystkie węzły z B będą przetworzone po m_b fazach. Rozważmy niepowiadomiony węzeł graniczny v . Niech X_v będzie jego zbiorem poprzedników należących do B . X_v jest niepusty. Weźmy j takie, że $2^{j-1} < |X_v| \leq 2^j$. Ponieważ zachodzi też $|Y_j| < 2^j$, S^j zawiera zbiór S_k^j , który wybiera X_v i omija Y_j . Zbiór ten będzie zbiorem nadawców w j -tej rundzie jednej z faz $i, i+1, \dots, i+m_j-1$. v otrzyma m , ponieważ
 - (a) dokładnie jeden poprzednik węzła v z X_v nada m w tej rundzie, bo S_k^j wybiera X_v
 - (b) żaden poprzednik węzła v z Y_j nie będzie nadawał, bo S_k^j omija Y_j .
 - (c) węzły uśpione tuż przed rozpoczęciem fazy i , spoza Y_j pozostają uśpione aż do fazy $i+m_j-1$ włącznie, z definicji Y_j

Dla każdego rozważanego v zachodzi $X_v \subseteq B$, więc $|X_v| \leq |B| \leq 2^b$. Stąd dla każdego v zachodzi $j \leq b$. Zatem w ciągu m_b faz postęp wynosi co najmniej $|B| > 2^{b-1}$, czyli zamortyzowany postęp wynosi $\Omega\left(\frac{2^b}{m_b}\right)$.

□

Jako bezpośredni wniosek dostajemy

Twierdzenie 10. *Jeśli w Protokole DO-BROADCAST wykonywanym na grafie o n węzłach użyjemy $(n, 2^j)$ -selektorów rozmiaru m_j , jego czas powiadomienia wyniesie $\mathcal{O}\left(n \cdot \max_j \frac{2^j}{m_j}\right)$.*

Warto zauważyć, że j w powyższym twierdzeniu może wynieść najwyżej $\lceil \log n \rceil$. Dla nas jednak najważniejszy będzie poniższy wniosek.

Wniosek 1. *Protokół DO-BROADCAST, gdy użyć w nim $(n, 2^j)$ -selektorów rozmiaru $\mathcal{O}(2^j \log^c n)$ osiąga czas powiadomienia $\mathcal{O}(n \log^{1+c} n)$.*

Nieprzypadkowo w Twierdzeniu 10 i Wniosku 1 mowa jest jedynie o czasie powiadomienia. Wydaje się, że trudno uzyskać terminację Protokołu DO-BROADCAST dotychczasowymi metodami — bo kiedy węzeł może przestać nadawać? W analizie rozważaliśmy dwa przypadki, w zależności od tego, czy istniał zbiór Y_j o mocy co najmniej 2^j . Jeśli taki zbiór nie istniał, powiadomione węzły brzegowe (a to one właśnie powinny umieć stwierdzić, kiedy stają się przetworzone) przekazywały w ciągu m_b faz wiadomość m do wszystkich swoich następników.

Wiemy co prawda, że $m_b = \mathcal{O}(n \log n)$, ale kłopoty sprawia przypadek, w którym istnieje zbiór Y_j o dużej mocy. Wtedy wiemy tylko, w ciągu m_j faz co najmniej 2^j nowych węzłów odebrało m . Nie wiemy za to, które są to węzły! W szczególności nie ma gwarancji, że będą to wszyscy sąsiedzi obecnych węzłów brzegowych!

Zdawałoby się, że skoro czas powiadomienia wynosi $\mathcal{O}(n \log^2 n)$, wystarczy, by każdy z węzłów nadawał zgodnie z protokołem tylko przez $\Theta(n \log^2 n)$ rund, dla odpowiednio dobranych stałych ukrytych w notacji. Czas terminacji byłby tego samego rzędu co czas powiadomienia. Niestety, tylko przy znanym n . Ponieważ n nie jest znane, węzły nadawałyby sygnały w nieskończoność według schematu podważania. A jednak da się zapewnić terminację Protokołu DO-BROADCAST w czasie równym czasowi powiadomienia — innymi metodami!

Opiszemy je wkrótce. Teraz wspomnimy tylko, że w [KP03b] podano protokół, którego czas powiadomienia zależy również od D . Wynosi on $\mathcal{O}(n \log n \log D)$, a

więc protokół z [KP03b] jest szybszy niż DO-BROADCAST, o ile $\log D = o(\log n)$. Jednak protokół z [KP03b] bazuje na znajomości n a pozbycie się tej zależności jest nietrywialne. Czas powiadomienia dla nieznanego n powiększa się o czynnik rzędu $\log \log n$ i wynosi $\mathcal{O}(n \log n \log \log n \log D)$.

4.3. Przeplatanie protokołów i szybsza terminacja. Protokół ROUND-ROBIN, jakkolwiek wolny, ma tę zaletę, że każdy z węzłów nadaje w nim sygnał dokładnie raz. Pokażemy jak wykorzystać ten protokół do uzyskania terminacji innych protokołów, lub poprawienia ich czasów terminacji. Podkreślmy, że nie jest to problem wydumany: nie potrafiliśmy dotychczas zapewnić terminacji protokołu DO-BROADCAST, zaś w protokołach BROAD-B i BROAD-C czas terminacji różnił się od czasu powiadomienia składnikiem, którego nie można zaniedbać.

Pomocny okazuje się pomysł, który pojawił się właśnie w tych protokołach, tj. równoległe wywołania różnych protokołów. Wystarczy przepleść dowolny protokół z Protokołem ROUND-ROBIN, np. w rundach nieparzystych wykonywać ROUND-ROBIN a w parzystych protokół, który chcemy poprawić. Jak poprzednio, wiedza węzłów w tych wywołaniach jest wspólna. Wspólna też będzie terminacja — przypominamy, że w omawianych protokołach węzeł dezaktywuje się, gdy jest pewien, że przekazał m do swoich następników. Wystarczy, że uda się to jednemu z przeplatanych protokołów, by w obu węzeł mógł się zdezaktywować.

Faktycznie korzystamy z własności terminacji Protokołu ROUND-ROBIN (dla znanego n), która gwarantuje, że każdy węzeł dezaktywuje się w czasie $\mathcal{O}(n)$ od otrzymania wiadomości m . Oczywiście jako terminatora można użyć dowolnego terminującego protokołu, ale liniowa względem n gwarancja ROUND-ROBIN jest najlepsza z tych, jakie uzyskaliśmy.

Przez przeplecenie protokołu DO-BROADCAST z ROUND-ROBIN dostajemy więc protokół o czasie terminacji $\mathcal{O}(n \log^2 n + n) = \mathcal{O}(n \log^2 n)$. Podobnie w przypadku Protokołu BROAD-B (w którym zakładamy znajomość n) dodatkowy składnik w czasie terminacji można zredukować z $\mathcal{O}(n \log n)$ do $\mathcal{O}(n)$. W protokole BROAD-C przeplatamy już zmodyfikowane Protokoły BROAD-B — w wyniku dodatkowy składnik w czasie terminacji zmniejsza się o czynnik logarytmiczny. Poniżej prezentujemy te fakty w postaci wniosków.

Wniosek 2. *Jeśli w protokole DO-BROADCAST wykonywanym na grafie o n węzłach użyjemy $(n, 2^j)$ -selektorów rozmiaru m_j i dodatkowo przepleciemy go z Protokołem ROUND-ROBIN, czas powiadomienia i terminacji wyniesie $\mathcal{O}\left(n \cdot \max_j \frac{2^j}{m_j}\right)$.*

Wniosek 3. *Protokół DO-BROADCAST przepleciony z Protokołem ROUND-ROBIN, gdy użyjemy w nim $(n, 2^j)$ -selektorów rozmiaru $\mathcal{O}(2^j \log^c n)$ mamy $m_j = \mathcal{O}(2^j \log^c n)$ osiąga czas powiadomienia i terminacji $\mathcal{O}(n \log^{1+c} n)$.*

Wniosek 4. *Protokół BROAD-B(n) przepleciony z Protokołem ROUND-ROBIN i uruchomiony na dowolnym grafie o n wierzchołkach, maksymalnym stopniu wejściowym d i maksymalnej odległości dowolnego wierzchołka od źródła D przeprowadza rozgłaszanie w czasie $\mathcal{O}(Dd \log n \log \frac{n}{d})$ i terminuje w czasie $\mathcal{O}(Dd \log n \log \frac{n}{d} + n)$.*

Wniosek 5. *Protokół BROAD-C oparty o przeplecione protokoły BROAD-B i ROUND-ROBIN, uruchomiony na dowolnym grafie o n wierzchołkach, maksymalnym stopniu wejściowym d i maksymalnej odległości dowolnego wierzchołka od źródła D przeprowadza rozgłaszanie w czasie $\mathcal{O}(Dd \log \frac{n}{d} \log^3 n)$ oraz terminuje w czasie $\mathcal{O}(Dd \log \frac{n}{d} \log^3 n + n \log^2 n)$.*

5. DOLNE OGRANICZENIA NA ROZMIAR RODZIN SELEKTYWNYCH, RODZIN SILNIE SELEKTYWNYCH I SELEKTORÓW

Zacznijemy od podania, bez dowodu, ograniczenia na rozmiar rodzin silnie selektywnych, pochodzącego z [CMS01].

Twierdzenie 11. *Niech \mathcal{F} będzie rodziną silnie (n, d) -selektywną. Wtedy*

- *Jeśli $2 \leq d \leq \sqrt{2n} - 1$, to $|\mathcal{F}| \geq \frac{d^2}{16 \log d} \log n$.*
- *Jeśli $d \geq \sqrt{2n}$, to $|\mathcal{F}| \geq n$.*

Jak wspomnieliśmy wcześniej, jest to ograniczenie na tyle duże, że z naszego punktu widzenia nie warto zajmować się rodzinami silnie selektywnymi. Do naszych zastosowań wystarczają rodziny selektywne i selektory, dla których znane są konstrukcje o mniejszym rozmiarze.

Dążymy do wykazania, że każda rodzina (n, d) -selektywna jest rozmiaru $\Omega(d \log(\frac{n}{d}))$. Korzystać będziemy z własności innej struktury kombinatorycznej, mianowicie *rodzin (słabo) rozłącznych (intersection free families)*. Są one dobrze znane i opisane w [FF85].

Definicja 4. *Dla $l \leq k \leq n$, rodzina \mathcal{G} k -podzbiorów zbioru $[n]$ jest l -rozłączna, jeśli*

$$\forall F_1, F_2 \in \mathcal{G} |F_1 \cap F_2| \neq l$$

W skrócie l -rozłączną rodzinę k -podzbiorów zbioru $[n]$ nazywać będziemy rodziną (n, k, l) -rozłączną. Z [FF85] pochodzi również poniższe twierdzenie, które przytaczamy bez dowodu.

Twierdzenie 12. *Niech \mathcal{G} będzie rodziną (n, k, l) -rozłączną, gdzie $2l + 1 \geq k$ oraz $k - l$ jest potęgą liczby pierwszej. Zachodzi wtedy*

$$|\mathcal{G}| \leq \binom{n}{l} \frac{\binom{2k-l-1}{k}}{\binom{2k-l-1}{l}}$$

Nas interesować będą rodziny $(n, k, k/2)$ -rozłączne, z pewnymi warunkami nałożonymi na n i k . Z powyższego twierdzenia wyprowadzimy dla nich prostsze ograniczenie górne na rozmiar.

Wniosek 6. *Niech \mathcal{G} będzie rodziną $(n, k, k/2)$ -rozłączną, gdzie k jest potęgą liczby 2 oraz $k \leq \frac{n}{26}$. Zachodzi wtedy*

$$\log |\mathcal{G}| \leq \frac{11k}{12} \log \frac{n}{k}$$

Dowód. Para $(k, l = k/2)$, gdzie k jest potęgą liczby 2 spełnia założenia Twierdzenia 12. Położmy $l = k/2$. Korzystać z poniższych prostych nierówności

$$\begin{aligned} \left(\frac{a}{b}\right)^b &\leq \binom{a}{b} \leq \left(\frac{a \cdot e}{b}\right)^b \\ \binom{a-1}{b} &= \frac{a-b}{a} \binom{a}{b} \end{aligned}$$

uzyskujemy

$$\begin{aligned} \log |\mathcal{G}| &\leq \log \left(\binom{n}{k/2} \frac{\binom{3k/2-1}{k}}{\binom{3k/2-1}{k/2}} \right) = \log \left(\frac{1}{2} \binom{n}{k/2} \frac{\binom{3k/2}{k}}{\binom{3k/2}{k/2}} \right) \leq \\ &\leq \log \left(\frac{1}{2} \left(\frac{2en}{k} \right)^{k/2} \left(\frac{3e}{2} \right)^k 3^{-k/2} \right) = \frac{k}{2} \log \frac{n}{k} + \frac{k}{2} \log 3 + \frac{3k}{2} \log e - \frac{k}{2} - 1 < \\ &< \frac{k}{2} \log \frac{n}{k} + \frac{5}{2}k \leq \frac{11k}{12} \log \frac{n}{k}, \end{aligned}$$

gdzie w ostatnim przejściu korzystamy z tego, że $k \leq \frac{n}{26}$, czyli $1 \leq \frac{1}{6} \log \frac{n}{k}$. \square

Teraz możemy sformułować i udowodnić twierdzenie o minimalnym rozmiarze rodziny (n, k) -selektywnej.

Twierdzenie 13. *Niech \mathcal{F} będzie rodziną (n, k) -selektywną, i niech $n > 2$, $2 \leq k \leq \frac{n}{64}$. Wtedy*

$$|\mathcal{F}| \geq \frac{k}{24} \log \frac{n}{k}$$

Dowód. Niech k' będzie potęgą liczby 2 taką, że $\frac{k}{2} \leq k' \leq k$. Zdefiniujmy graf G w następujący sposób: zbiór wierzchołków G to zbiór wszystkich k' -podzbiorów zbioru $[n]$, zaś krawędzie łączą wierzchołki mające dokładnie $\frac{k'}{2}$ wspólnych elementów. Niech $\chi(G)$ oznacza liczbę chromatyczną grafu G , zaś $\alpha(G)$ moc największego zbioru niezależnego w G . Wykażemy, że

$$\frac{k}{24} \log \frac{n}{k} \leq \log \chi(G) \leq |\mathcal{F}|.$$

Do wykazania lewej nierówności skorzystamy z trywialnego oszacowania $\chi(G) \geq \frac{|V(G)|}{\alpha(G)}$. Ponieważ wierzchołki dowolnego zbioru niezależnego grafu G tworzą rodzinę $(n, k', \frac{k'}{2})$ -niezależną, możemy skorzystać z Wniosku 6, by uzyskać

$$\begin{aligned} \log \chi(G) &\geq \log |V(G)| - \log \alpha(G) \geq \log \binom{n}{k'} - \frac{11k'}{12} \log \frac{n}{k} \geq \\ &\geq k' \log \frac{n}{k'} - \frac{11k'}{12} \log \frac{n}{k'} \geq \frac{k'}{12} \log \frac{n}{k'} \geq \frac{k}{24} \log \frac{n}{k}. \end{aligned}$$

Przy dowodzie prawej nierówności skorzystamy z równie trywialnej nierówności $\chi\left(\bigcup_{i=1}^t G_i\right) \leq \prod_{i=1}^t \chi(G_i)$, prawdziwej dla grafów o tym samym zbiorze wierzchołków. Niech $\mathcal{F} = \{F_1, F_2, \dots, F_{|\mathcal{F}|}\}$. Grafy G_i , dla $1 \leq i \leq |\mathcal{F}|$, definiujemy następująco: $V(G_i) = V(G)$ oraz $E(G_i) = \{\{u, v\} \in E(G) : |F_i \cap (u \div v)| = 1\}$. Każda krawędź $\{u, v\} \in E(G)$ należy do co najmniej jednego ze zbiorów $E(G_i)$, bo

- (1) $|u \div v| = k' \leq k$ oraz
- (2) \mathcal{F} jest rodziną (n, k) -selektywną.

W takim razie $G = \bigcup_{i=1}^{|\mathcal{F}|} G_i$. Ponadto grafy G_i są dwudzielne, bo jeden z wierzchołków u, v połączonych w G_i krawędzią ma nieparzystą liczbę wspólnych elementów z F_i , a drugi parzystą:

$$1 = |(u \div v) \cap F_i| = |u \cap F_i| + |v \cap F_i| - 2|u \cap v \cap F_i| \equiv |u \cap F_i| + |v \cap F_i| \pmod{2}.$$

Zachodzi więc poniższy ciąg nierówności:

$$\log \chi(G) = \log \chi\left(\bigcup_{i=1}^{|\mathcal{F}|} G_i\right) \leq \log \prod_{i=1}^{|\mathcal{F}|} \chi(G_i) = \sum_{i=1}^{|\mathcal{F}|} \log \chi(G_i) \leq |\mathcal{F}|.$$

\square

Ograniczenie górne z Twierdzenia 6 i dolne z Twierdzenia 13 są równe z dokładnością do stałych, czyli optymalny rozmiar rodziny (n, k) -selektywnej wynosi $\Theta(k \log \frac{n}{k})$.

W [Ind02] zaznaczono, że z Twierdzenia 13 i Faktu 2 można uzyskać dolne ograniczenie $\Omega(k \log \frac{n}{k})$ na rozmiar selektora. Twierdzenie 13 przytoczono błędnie jako dające ograniczenie dolne $\Omega(k \log n)$ na rozmiar rodziny selektywnej zamiast poprawnego $\Omega(k \log \frac{n}{k})$. Wówczas znane było jedynie słabsze górne ograniczenie

$\mathcal{O}(k \log n)$ z Twierdzenia 4 z [CMS01], a ograniczenie górne $\mathcal{O}(k \log \frac{n}{k})$ z Twierdzenia 6 z [CMS03] udowodniono później.

Niezależnie od tej nieścisłości, nie da się łatwo wysnuć dolnych ograniczeń na rozmiar selektorów dzięki takim ograniczeniom na rozmiar rodzin selektywnych. Podstawą wnioskowania miał być Fakt 2, mówiący, że suma odpowiednich selektorów jest rodziną selektywną. Jeśli założyć, że istnieją (n, k) -selektory rozmiaru $o(k \log \frac{n}{k})$, to z Faktów 2 i 4 wynika istnienie rodzin (n, k) -selektywnych rozmiaru $o(k \log \frac{n}{k})$, co jest sprzeczne z Twierdzeniem 13.

Sprzeczność ta nie daje ograniczenia dolnego $\Omega(k \log \frac{n}{k})$ dla dowolnych n, k , jak sugerowano w [Ind02]. Wynika z niej jedynie, że dla nieskończenie wielu par (n, k) rozmiar (n, k) -selektora wynosi $\Omega(k \log \frac{n}{k})$. O k z tych par można dodatkowo przyjąć, że są potęgami liczby 2.

W definicji 3 selektora mamy $\frac{k}{2} \leq |X| \leq k$, nie wynika więc z niej monotoniczność ze względu na k : (n, k) -selektor nie musi być (n, k') -selektorem dla $k' < k$. Co za tym idzie, nie możemy przenieść uzyskanego dla pewnych szczególnych par (n, k) ograniczenia dolnego na wszystkie pary (n, k) .

5.1. Dolne ograniczenia na czas rozgłaszania. Zastosujemy teraz Twierdzenie 13, mówiące, że dla rodziny (n, k) -selektywnej \mathcal{F} , $|\mathcal{F}| = \Omega(k \log \frac{n}{k})$, by uzyskać dolne ograniczenie $\Omega(n \log n)$ na czas deterministycznego rozgłaszania bez spontanicznej komunikacji w sieciach radiowych. Jakkolwiek w prezentowanych twierdzeniach pojawiają się dodatkowo parametry D i d , można tak je dobrać, by uzyskać ograniczenie $\Omega(n \log n)$. Dowody polegają na takim przypisaniu identyfikatorów węzłom odpowiedniego *pełnego grafu warstwowego*, że przekazanie m z dowolnej warstwy do kolejnej wymaga długiego czasu. Oba twierdzenia pochodzą z [CMS01].

Definicja 5. *Graf skierowany G nazywamy pełnym grafem warstwowym, gdy*

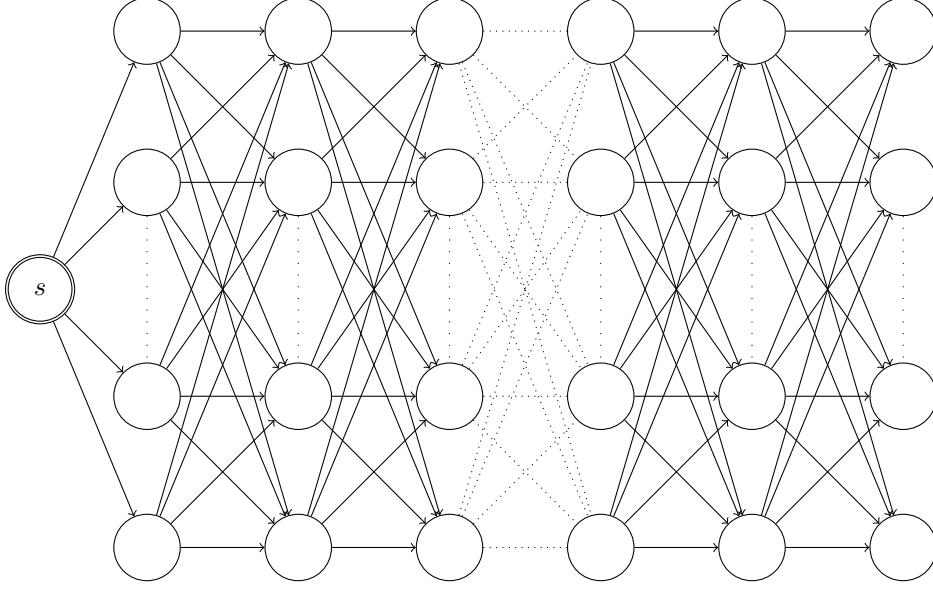
- (1) G składa się z $D + 1$ warstw, L_0, L_1, \dots, L_D dla pewnego D naturalnego,
- (2) L_i to zbiór węzłów odległych od źródła o i ,
- (3) $L_0 = \{s\}$, tj. zawiera wyłącznie źródło,
- (4) dla $0 \leq i < D$, (L_i, L_{i+1}) tworzy pełny graf dwudzielny z krawędziami skierowanymi od L_i do L_{i+1} .

Twierdzenie 14. *Dla dowolnego deterministycznego protokołu rozgłaszania P bez spontanicznej komunikacji oraz dowolnych n i D takich, że $D \geq 64$, istnieje n -wierzchołkowy graf G^P o maksymalnej odległości od źródła wynoszącej D taki, że czas powiadomienia P na G^P wynosi $\Omega(n \log D)$.*

Dowód. Twierdzenie udowodnimy dla silniejszych *protokołów półwszechwiedzących*. Protokół jest półwszechwiedzący, gdy akcja węzła v w chwili t zależy wyłącznie od

- jego identyfikatora v ,
- chwili t ,
- n , tj. liczby węzłów w grafie,
- d , tj. maksymalnego stopnia wejściowego węzłów,
- historii transmisji wszystkich węzłów grafu G_v , tj. zapisu odebranych przez niego sygnałów w czasie od 0 do t , gdzie G_v to *graf poprzedników v* : podgraf grafu G indukowany na wierzchołkach, z których v jest osiągalny.

Protokół półwszechwiedzący jest silniejszy od dotychczas rozważanych, bo v zna nie tylko swój stan, ale również stany wszystkich swoich poprzedników przez parametry grafu: n i d . W szczególności każdy węzeł od samego początku zna m , skoro jest osiągalny ze źródła. Wymagamy jednak, by m zostało mu przesłane przez inny węzeł a jednocześnie nie dopuszczamy spontanicznej komunikacji. Każdy z węzłów może więc nadawać dopiero gdy m zostanie do niego przesłane, tj. sama znajomość m nic mu nie daje.



RYSUNEK 2. Pełny graf warstwowy.

Teraz definiujemy graf G^P . G^P jest pełnym grafem warstwowym o $D + 1$ warstwach, L_0, L_1, \dots, L_D , jak w Definicji 5. Każda z warstw L_1, L_2, \dots, L_{D-1} ma nie więcej niż $\lceil \frac{n}{2D} \rceil$ węzłów, zaś L_D stanowią wszystkie pozostałe węzły. Kluczem dowodu jest takie rozmieszczenie węzłów w warstwach i przypisanie im identyfikatorów, że przekazanie m do każdej kolejnej warstwy trwa $\Omega(\frac{n}{D} \log D)$ rund.

Ponieważ $D \geq 64$, zachodzi $\lceil \frac{n}{2D} \rceil \leq \frac{1}{64} \lceil \frac{n}{2} \rceil$ i z Twierdzenia 13 wynika, że istnieje stała $c > 0$ taka, że każda rodzina $(\lceil \frac{n}{2} \rceil, \lceil \frac{n}{2D} \rceil)$ -selektywna ma rozmiar przynajmniej $T = \lceil c \frac{n}{D} \log D \rceil$. Dowodzone twierdzenie jest prostą konsekwencją poniższego lematu. \square

Lemat 2. *Dla każdego $j = 1, 2, \dots, D$, można rozmieścić węzły w warstwach L_1, L_2, \dots, L_j grafu G^P tak, że P nie przekaże wiadomości m do warstwy L_j przed chwilą $(j - 1)T$, gdzie $T = \lceil c \frac{n}{D} \log D \rceil$ dla pewnej stałej $c > 0$ jest dolnym ograniczeniem na rozmiar rodziny $(\lceil \frac{n}{2} \rceil, \lceil \frac{n}{2D} \rceil)$ -selektywnej.*

Dowód. Indukcja względem j . Teza jest spełniona w sposób trywialny dla $j = 1$. By dowieść kroku indukcyjnego zdefiniujmy dla $j = 1, 2, \dots, D - 1$

- G_j jako podgraf konstruowanego G^P indukowany na węzłach z warstw L_0, L_1, \dots, L_j ; G_j jest grafem poprzedników dla każdego węzła z warstwy L_{j+1}
- $\Delta T_j = \{(j - 1)T, (j - 1)T + 1, \dots, jT - 1\}$

Założmy, że teza zachodzi dla każdego $j' < j$ a warstwy L_1, L_2, \dots, L_{j-1} zostały już zgodnie z nią ustalone. Zdefiniujmy

$$R_j = V(G^P) \setminus \bigcup_{h=0}^{j-1} L_h,$$

tj. R_j to zbiór węzłów, które nie zostały jeszcze przydzielone do żadnej warstwy. Spośród nich wybierzemy te, które utworzą warstwę L_j . Możemy wybierać spośród

wielu węzłów: $|R_j| \geq \lceil \frac{n}{2} \rceil$, bo

$$|R_j| = n - \sum_{h=0}^{j-1} |L_h| \geq n - \left\lceil \frac{n}{2D} \right\rceil (D-2) - 1 \geq \left(n - \left\lceil \frac{n}{2} \right\rceil \right) + 2 \left\lceil \frac{n}{2D} \right\rceil - 1 \geq \left\lceil \frac{n}{2} \right\rceil.$$

Niech więc R'_j będzie dowolnym podzbiorem R_j o mocy $\lceil \frac{n}{2} \rceil$. Symulujemy zachowanie węzłów z R'_j podczas wykonania protokołu P przy założeniu, że G_{j-1} jest grafem poprzedników każdego z nich. Oznaczmy

$$S_t = \{v \in R'_j : i \text{ nadaje w chwili } (j-1)T + t\}$$

$$\mathcal{F} = \{S_1, S_2, \dots, S_{T-1}\}.$$

Ponieważ $|\mathcal{F}| < T$, \mathcal{F} nie jest rodziną $(\lceil \frac{n}{2} \rceil, \lfloor \frac{n}{2D} \rfloor)$ -selektywną, więc istnieje podzbiór $L \subseteq R'_j$ o mocy nie większej niż $\lfloor \frac{n}{2D} \rfloor$, którego nie wybiera żaden ze zbiorów z rodziny \mathcal{F} . Wybieramy $L_j := L$, gwarantując, że węzły z warstwy L_j nie przekażą wiadomości m do L_{j+1} aż do ostatniej rundy przedziału ΔT_j włącznie — pamiętajmy, że same poznają m najwcześniej w rundzie $(j-1)T$. \square

Twierdzenie 15. *Dla dowolnego deterministycznego protokołu rozgłaszania P bez spontanicznej komunikacji oraz n , D i d takich, że $(D-1)d+1 \leq \frac{n}{2}$ i jednocześnie $d \leq \frac{n}{128}$ istnieje n -wierzchołkowy graf G^P o maksymalnej odległości od źródła wynoszącej D i maksymalnym stopniu wejściowym d taki, że czas powiadomienia P na G^P wynosi $\Omega(Dd \log \frac{n}{d})$.*

Dowód. Dowód jest podobny do dowodu Twierdzenia 14, więc przedstawiamy tylko główne różnice. Teraz każda z warstw L_1, L_2, \dots, L_{D-1} zawiera najwyżej d węzłów, zaś L_D (jak poprzednio) składa się ze wszystkich pozostałych. L_D można połączyć z L_{D-1} tak, by stopnie wejściowe węzłów z L_D nie przekraczały d , bo stopień wyjściowy węzłów z L_{D-1} może być dowolnie duży. Nie gwarantujemy, że pokonanie ostatniej warstwy wymaga dużego czasu — przez ograniczony stopień wejściowy grafy poprzedników węzłów z L_D mogą nie zawierać wszystkich węzłów z L_{D-1} .

Czas na pokonanie pozostałych warstw to $\Omega(d \log \frac{n}{d})$, minimalny rozmiar rodziny $(n/2, d)$ -selektywnej. Ograniczenie to, dzięki nierówności $d \leq \frac{n}{128}$, wynika z Twierdzenia 13. Z kolei warunek $(D-1)d+1 \leq \frac{n}{2}$ gwarantuje, że zawsze jest co najmniej $\frac{n}{2}$ wierzchołków, z których wybrać można te do kolejnej warstwy. \square

Oczywiście można tak dobrać D i d , by Twierdzenia 14 i 15 dawały dolne ograniczenie na czas deterministycznego rozgłaszania $\Omega(n \log n)$. Za to ograniczenia wyrażone z użyciem parametrów D i d są ogólniejsze. Założenia obu twierdzeń w [CMS01] miały drobne błędy. My przedstawiliśmy wersje poprawione. Ograniczenia z [CMS01] są dużo silniejsze dla małych D od ograniczeń znanych wcześniej, np. $\Omega(D \log n)$ z [CGGPR00].

5.2. Protokół rozgłaszania w grafach warstwowych o optymalnym czasie powiadomienia. Istnieje wiele dowodów ograniczenia dolnego $\Omega(n \log n)$ na czas deterministycznego rozgłaszania w sieciach radiowych — choćby wspomniane ograniczenie $\Omega(D \log n)$ z [CGGPR00] dla $D = \Theta(n)$. Wybraliśmy to z [CMS01], bo pokazuje ścisły związek problemu deterministycznego rozgłaszania w sieciach radiowych z rodzinami selektywnymi.

Za to wszystkie te dowody są do siebie podobne — polegają na konstrukcji trudnego dla danego protokołu pełnego grafu warstwowego. Zachowanie protokołów na takich grafach jest łatwo analizować — nie zawierają cykli, a dodatkowo wszystkie wierzchołki z jednej warstwy poznają m w tej samej chwili.

Dla pełnych grafów warstwowych istnieje optymalny protokół rozgłaszania, tj. protokół mający czas powiadomienia $\mathcal{O}(n \log n)$. Więc by uzyskać lepsze dolne

ograniczenie na czas rozgłaszania, trzeba konstruować inne, zapewne trudniejsze do analizy grafy. Wspomniany protokół — Protokół CLN z [CGR00] — prezentujemy poniżej. Podkreślamy jednak, że jego czas powiadomienia jest wyrażony wyłącznie przez n , więc być może nie jest optymalny, gdy $D = o(n)$.

Dla $j = 0, 1, \dots, \lceil \log n \rceil$, niech $\mathcal{S}^j = (S_0^j, S_1^j, \dots, S_{m_j-1}^j)$ będzie $(n, 2^j)$ -selektorem, zaś m_j jego rozmiarem. Z twierdzenia 5 wiemy, że istnieją takie selektory, że $m_j = \mathcal{O}(2^j \log n)$. Niech

$$\bar{\mathcal{S}} = (S_0^0, S_1^0, \dots, S_{m_0-1}^0, S_0^1, S_1^1, \dots, S_{m_1-1}^1, \dots, S_{m_{\lceil \log n \rceil}-1}^{\lceil \log n \rceil}),$$

tj. $\bar{\mathcal{S}}$ jest ciągiem zbiorów ze wszystkich selektorów, w porządku rosnącego parametru k . Gdy tylko węzeł v otrzyma m , zaczyna nadawać zgodnie z ciągiem $\bar{\mathcal{S}}$, tj. jeśli otrzymał m w chwili t_v , w chwili $t_v + t$ nadaje wtedy i tylko wtedy gdy znajduje się w t -tym zbiorze ciągu $\bar{\mathcal{S}}$.

Dane niejawne: $G(V, E)$	
Dane jawne: n, s, m	
Dane pomocnicze: $(n, 2^k)$ -selektory $\mathcal{S}^k = (S_0^k, S_1^k, \dots, S_{m_k-1}^k)$, dla $0 \leq k \leq \lceil \log n \rceil$ oraz ciąg $\bar{\mathcal{S}} = (S_0^0, S_1^0, \dots, S_{m_0-1}^0, S_0^1, S_1^1, \dots, S_{m_1-1}^1, \dots, S_{m_{\lceil \log n \rceil}-1}^{\lceil \log n \rceil})$	
/* Przez $\mathcal{S}_{(j)}$ oznaczać będziemy j -ty element ciągu $\bar{\mathcal{S}}$	*/
/* pierwsza runda	*/
s nadaje m ;	
for $i = 2, 3, \dots$ do /* kolejne rundy	*/
foreach $v \neq s$ do	
if v otrzymał m po raz pierwszy w rundzie $i - t$ oraz $v \in \mathcal{S}_{(t)}$ dla $t \in [\bar{\mathcal{S}}]$ then	
v nadaje m ;	
/* Co jaśniej można opisać tak:	*/
/* Węzeł $v \neq s$, który odebrał m po raz pierwszy w rundzie t_v , od następnej rundy nadaje m zgodnie z ciągiem $\bar{\mathcal{S}}$,	*/
/* tj. w rundzie $t_v + t$ nadaje wtedy i tylko wtedy gdy znajduje się w t -tym zbiorze ciągu $\bar{\mathcal{S}}$	*/
/* jak widać, v dezaktywuje się po rundzie $t_v + \bar{\mathcal{S}} $	*/

Protokół 9: Protokół CLN

Twierdzenie 16. *Protokół CLN przeprowadza rozgłaszanie bez użycia spontanicznej komunikacji i terminuje w czasie $\mathcal{O}(n \log n)$ dla dowolnego pełnego grafu warstwowego o n wierzchołkach.*

Dowód. Węzły nadają momentalnie po otrzymaniu m , bez względu na numer rundy. Z definicji pełnego grafu warstwowego wynika, że wszystkie węzły jednej warstwy poznają m w tej samej chwili, czyli de facto są zsynchronizowane. Jeśli $|L_i| = w_i$, przesłanie wiadomości do warstwy L_{i+1} gwarantuje selektor $\mathcal{S}^{\lceil \log w_i \rceil}$. Jednak

$$(5) \quad \sum_{j=0}^l 2^j \log n \leq 2 \cdot 2^l \log n,$$

więc czas potrzebny na przesłanie wiadomości przez L_i do L_{i+1} wynosi $\mathcal{O}(w_i \log n)$. Sumując po warstwach otrzymujemy czas powiadomienia $\mathcal{O}(n \log n)$.

Węzeł v dezaktywuje się po $|\bar{\mathcal{S}}|$ rundach od otrzymania m .

$$|\bar{\mathcal{S}}| = \sum_{k=0}^{\lceil \log n \rceil} |\mathcal{S}^k| = \sum_{k=0}^{\lceil \log n \rceil} \mathcal{O}(2^k \log n) = \mathcal{O}(n \log n) ,$$

więc czas terminacji również wynosi $\mathcal{O}(n \log n)$. \square

Zauważmy, że nie korzystaliśmy z własności selektorów dotyczącej „omijania” zbiorów Y . Wobec tego zamiast selektorów można w Protokole CLN użyć rodzin selektywnych, dla których Twierdzenie 6 gwarantuje lepsze ograniczenie górne na rozmiar. Równie dobre ograniczenia na rozmiar selektorów uzyskamy w następnym rozdziale.

W [KP03b] autorzy zaprezentowali protokół, który korzystając z silniejszego ograniczenia na rozmiar rodzin selektywnych, $\mathcal{O}(k \log \frac{n}{k})$, uzyskuje czas powiadomienia $\mathcal{O}(n \log D)$, czyli na mocy twierdzenia 14 jest optymalny. Niestety, taki czas osiągnąć jest jedynie gdy znamy n . Jeśli n jest nieznanne, przez to, że D może wynosić nawet $n - 1$, czas nadal wynosi $\mathcal{O}(n \log n)$.

Poniżej pokazujemy w oparciu o silniejsze ograniczenia na rozmiar rodzin selektywnych, że

- węzeł v dezaktywuje się po $\mathcal{O}(n)$ rundach od otrzymania m oraz
- wystarczy poprawić analizę Protokołu CLN, by dostać czas powiadomienia $\mathcal{O}(n \log D)$ — analizę wzorujemy na analizie z [KP03b].

Twierdzenie 17. *W Protokole CLN każdy węzeł dezaktywuje się po $\mathcal{O}(n)$ rundach od otrzymania m .*

Dowód. Węzeł v dezaktywuje się po $|\bar{\mathcal{S}}|$ rundach od otrzymania m . Tym razem przy szacowaniu $|\bar{\mathcal{S}}|$ korzystamy z Twierdzenia 6 oraz Faktu 4:

$$|\bar{\mathcal{S}}| = \sum_{k=0}^{\lceil \log n \rceil} |\mathcal{S}^k| = \sum_{k=0}^{\lceil \log n \rceil} \mathcal{O}\left(2^k \log \frac{n}{k}\right) = \mathcal{O}(n) .$$

\square

Twierdzenie 18. *Protokół CLN przeprowadza rozgłaszanie bez użycia spontanicznej komunikacji i terminuje w czasie $\mathcal{O}(n \log D)$ dla dowolnego pełnego grafu warstwowego o n wierzchołkach i maksymalnej odległości od źródła D oraz znanym n .*

Dowód. Selektory zastępujemy rodzinami selektywnymi i z Twierdzenia 6 wiemy, że $m_j = \mathcal{O}\left(2^j \log \frac{n}{2^j}\right)$. Pozwala to to lepiej oszacować czas potrzebny na przekazanie wiadomości z warstwy L_i do warstwy L_{i+1} — z Faktu 4 wynika, że nierówność (5) możemy wzmocnić do

$$\sum_{j=0}^l 2^j \log \frac{n}{2^j} = \mathcal{O}\left(2^l \log \frac{n}{2^l}\right) ,$$

co daje $t_i = \mathcal{O}\left(w_i \log \frac{n}{w_i}\right)$, gdzie t_i to czas potrzebny warstwie L_i (o mocy w_i) na przekazanie wiadomości do L_{i+1} . Teraz oszacujemy sumę czasów t_i dla wszystkich warstw poza ostatnią.

$$(6) \quad \sum_{i=0}^{D-1} t_i = \mathcal{O}\left(\sum_{i=0}^{D-1} w_i \log \frac{n}{w_i}\right) = \mathcal{O}\left(\sum_{i=0}^{D-1} \log \frac{n^{w_i}}{w_i^{w_i}}\right) = \mathcal{O}\left(\log \frac{n^{n-w_D}}{\prod_{i=0}^{D-1} w_i^{w_i}}\right) .$$

Gdy na iloczyn w mianowniku logarytmu spojrzymy jak na iloczyn $n - w_D$ czynników, z nierówności między średnimi geometryczną i harmoniczną dostaniemy

$$(7) \quad \prod_{i=0}^{D-1} w_i^{w_i} \geq \left(\frac{n - w_D}{\sum_{i=0}^{D-1} w_i \cdot \frac{1}{w_i}} \right)^{n - w_D} = \left(\frac{n - w_D}{D} \right)^{n - w_D}.$$

Wstawiając (7) w (6) dostajemy

$$\sum_{i=0}^{D-1} t_i = \mathcal{O} \left((n - w_D) \log \frac{nD}{n - w_D} \right) = \mathcal{O}(n \log D),$$

ponieważ funkcja $f(x) = x \log \frac{c}{x}$, gdzie c to dodatnia stała, jest rosnąca dla $x < \frac{c}{e}$:

$$f'(x) = \log \frac{c}{x} + x \left(-\frac{c}{x^2} \cdot \frac{\log e}{\frac{c}{x}} \right) = \log \frac{c}{x} - \log e.$$

U nas $c = nD$, więc dla $D \geq 3$ zachodzi $\frac{c}{e} > n$. Oczywiście zachodzi również $x = n - w_D < n$, więc dla $D \geq 3$ mamy

$$(n - w_D) \log \frac{nD}{n - w_D} < n \log \frac{nD}{n} = n \log D.$$

Jeśli $D < 3$, prosta analiza trywialnych przypadków wykazuje, że twierdzenie również zachodzi:

- $D = 0 \rightarrow$ źródło jest jedynym wierzchołkiem grafu
- $D = 1 \rightarrow$ w pierwszej rundzie źródło powiadamia wszystkie węzły
- $D = 2 \rightarrow$ czas powiadomienia wynosi $t_0 + t_1 = 1 + \mathcal{O} \left(w_i \log \frac{n}{w_i} \right)$. Z analizy funkcji f wiemy, że maksimum osiągnięte jest dla $w_i = \frac{n}{e}$, a wtedy $t_0 + t_1 = \mathcal{O}(n)$.

Terminację w czasie $\mathcal{O}(n \log D)$ gwarantuje Twierdzenie 17. \square

6. KILKA SŁÓW O UOGÓLNIONYCH SELEKTORACH

W [BGV03] autorzy nazywają selektorami stuktury nieco inne od omawianych przez nas. Ponieważ są w pewnym sensie uogólnieniem selektorów z [CGR00], by uniknąć niejasności, będziemy je nazywać *uogólnionymi selektorami*. Znane dla nich dolne i górne ograniczenia na rozmiar dają ściśle dolne i górne ograniczenie $\Theta(k \log \frac{n}{k})$ dla (n, k) -selektorów — dokładnie takie, jak dla rodzin selektywnych.

6.1. Uogólnione selektory oraz ich związki z poprzednimi strukturami.

Definicja 6. Dla $1 \leq r \leq k \leq n$, *uogólnionym (n, k, r) -selektorem nazywamy rodzinę \mathcal{S} podzbiorów zbioru $[n]$ taką, że*

$$\forall X \subseteq [n], |X| = k \exists S_1, S_2, \dots, S_r \in \mathcal{S} \forall i \in [r] |X \cap S_i| = 1 \wedge \forall i, j \in [r], i \neq j S_i \cap S_j \cap X = \emptyset$$

Główna różnica w stosunku do selektorów oraz rodzin selektywnych polega na tym, że zbiór X ma moc dokładnie k , a ponadto parametr r określa, ile różnych elementów z takiego zbioru X można wybrać. W [BGV03] wskazano związki uogólnionych selektorów z rodzinami selektywnymi, silnie selektywnymi oraz selektorami. Obserwacje te były nieściśle: pisano o „odpowiadaniu”, które raz oznaczało równoważność, raz implikację w jedną stronę a raz w drugą. Poniżej prezentujemy dokładne związki wspomnianych struktur. Wszystkie fakty są prostymi konsekwencjami definicji.

Fakt 5. *Każdy uogólniony (n, k, k) -selektor jest rodziną silnie (n, k) -selektywną i odwrotnie.*

Fakt 6. *Każdy uogólniony $(n, 2k, \frac{3}{2}k)$ -selektor jest (n, k) -selektorem.*

Fakt 7. *Każdy (n, k) -sektor jest uogólnionym $(n, 2k, k + 1)$ -sektorem.*

Fakt 8. *Każda rodzina (n, k) -selektorywna jest uogólnionym $(n, k, 1)$ -sektorem.*

Fakt 9. *Suma $\bigcup_{i=0}^{\lceil \log k \rceil} \mathcal{S}_i$, gdzie \mathcal{S}_i dla $i \neq 0$ jest dowolnym uogólnionym $(n, 2^i, 2^{i-1})$ -sektorem a \mathcal{S}_0 dowolnym uogólnionym $(n, 1, 1)$ -sektorem, jest rodziną (n, k) -selektorywną.*

Zauważmy podobieństwo Faktów 9 i 2. Fakt 9 bierze się z analogicznego spostrzeżenia, że uogólniony $(n, 2^i, 2^{i-1})$ -sektor zachowuje się jak rodzina $(n, 2^i)$ -selektorywna, o ile zbiór X poza nierównością $|X| \leq 2^i$ spełnia także $|X| > 2^{i-1}$. Dla $i = 0$, tj. dla X -ów singletonów, należy użyć uogólnionego $(n, 1, 1)$ -sektora. Uogólnionym $(n, 1, 1)$ -sektorem, jest np. $\{[n]\}$. Podkreślmy, że Fakt 9 jest podobny do Faktu 2, jednak pojawiają się w nim selektory o mniejszych parametrach, niż gdyby bezpośrednio skorzystać z Faktów 2 i 6.

W [BGV03] napisane jest, że uogólnione (n, k, k) -selektory odpowiadają rodzinom silnie (n, k) -selektorywnym, co stwierdzamy w Fakcie 5. Dla rodzin selektorywnych i selektorów, każda z opisanych w [BGV03] odpowiedniości zachodzi tylko w jedną stronę, w dodatku nie tę samą. Według [BGV03] uogólniony $(n, 2k, \frac{3}{2}k)$ -sektor odpowiada (n, k) -sektorowi uogólniony $(n, k, 1)$ -sektor odpowiada rodzinie (n, k) -selektorywnej. Warto porównać te stwierdzenia z Faktami 6 i 7 dla selektorów oraz 8 i 9 dla rodzin selektorywnych.

6.2. Ograniczenia na rozmiar uogólnionych selektorów. Mimo nieściślych odpowiedzi, uogólnione selektory pozwolą nam zamknąć kwestię rozmiaru selektorów, dzięki Faktom 6 i 7 oraz górnym i dolnym ograniczeniom na rozmiar uogólnionych selektorów. Pochodzące z [BGV03] ograniczenia przytaczamy bez dowodów.

Twierdzenie 19. *Dla każdych $k, n, r \in \mathbb{N}$, $1 \leq r \leq k < n$, istnieje uogólniony (n, k, r) -sektor o rozmiarze t , gdzie*

$$t < \frac{ek^2}{k-r+1} \ln \frac{n}{k} + \frac{ek(2k-1)}{k-r+1}$$

Twierdzenie 20. *Dla każdych $k, n, r \in \mathbb{N}$, $1 \leq r \leq k < n$, minimalny rozmiar uogólnionego (n, k, r) -sektora, $t(n, k, r)$ spełnia nierówność*

$$t(n, k, r) \geq \log \binom{n}{r-1} - k + 1 \geq (r-1) \log \frac{n}{r-1} - k + 1$$

W [BGV03] podano także silniejsze ograniczenie dolne, zachodzące gdy $k < 2r - 2$. Z Faktu 7 wynika, że (n, k) -sektor jest uogólnionym $(n, 2k, k + 1)$ -sektorem. Można użyć dla niego silniejszego ograniczenia dla uogólnionego $(n, 2k, k)$ -sektora (którym oczywiście jest), ale nie jest to konieczne. Skorzystanie z powyższych dwóch twierdzeń i związków między zwykłymi i uogólnionymi selektorami, daje równość (z dokładnością do stałej) dolnych i górnych ograniczeń na rozmiar (n, k) -sektora. Wynoszą one $\Theta(k \log \frac{n}{k})$.

6.3. Ograniczenia na rozmiar selektorów.

Twierdzenie 21. *Dla każdych $n, k \in \mathbb{N}$, $1 \leq k \leq n$, istnieje (n, k) -sektor rozmiaru $\mathcal{O}(k \log \frac{n}{k})$.*

Dowód. Z Faktu 6 wynika, że dowolny uogólniony $(n, 2k, \frac{3}{2}k)$ -sektor jest (n, k) -sektorem, zaś Twierdzenie 19 daje ograniczenie górne na rozmiar uogólnionych (n, k, r) -sektorów. Podstawiając $2k$ za k i $\frac{3}{2}k$ za r stwierdzamy istnienie (n, k) -sektora o rozmiarze

$$t \leq \frac{4ek^2}{2k - \frac{3}{2}k + 1} \ln \frac{n}{2k} + \frac{2ek(4k-1)}{2k - \frac{3}{2}k + 1} < 8ek \ln \frac{n}{2k} + 16ek = \mathcal{O}\left(k \log \frac{n}{k}\right).$$

Skoro $r = \frac{3}{2}k$, wymagamy, by k było parzyste, jednak dla k nieparzystego kładzimy $r = \frac{3k+1}{2}$, nie zwiększając asymptotycznego rozmiaru. \square

Twierdzenie 22. *Dla każdych $n, k \in \mathbb{N}$, dowolnej stałej $\epsilon > 0$, $1 \leq k \leq \frac{n}{4+\epsilon}$, dowolny (n, k) -selektor ma rozmiar $\Omega\left(k \log \frac{n}{k}\right)$.*

Dowód. Z Faktu 7 wynika, że dowolny (n, k) -selektor jest uogólnionym $(n, 2k, k+1)$ -selektorem, zaś Twierdzenie 20 daje ograniczenie dolne na rozmiar uogólnionych (n, k, r) -selektorów. Podstawiając $2k$ za k i $k+1$ za r otrzymujemy górne ograniczenie na rozmiar dowolnego (n, k) -selektora \mathcal{S} :

$$|\mathcal{S}| \geq k \log \frac{n}{k} - 2k + 1 = \Omega\left(k \log \frac{n}{k}\right).$$

Warunek $k \leq \frac{n}{4+\epsilon}$ gwarantuje, że $k \log \frac{n}{k}$ jest istotnie większe od $2k$. \square

7. KONSTRUKCJA SELEKTORÓW

7.1. Wstęp do konstrukcji. Dzięki Twierdzeniom 6, 13, 21, 22 wiemy, że istnieją rodziny (n, k) -selektywne oraz (n, k) -selektory o asymptotycznie optymalnym rozmiarze $\mathcal{O}\left(k \log \frac{n}{k}\right)$. Dowody Twierzeń 6 i 5 pokazują nawet, jak można wylosować te stuktury z dużym prawdopodobieństwem sukcesu, uzyskując optymalny rozmiar dla rodzin selektywnych i bliski optymalnemu dla selektorów.

Niestety wciąż nie umiemy skonstruować ich deterministycznie w rozsądnym czasie. Najlepiej, by był to czas wielomianowy. Przypominamy, że w najszybszych prezentowanych protokołach węzły korzystały z selektorów lub rodzin selektywnych, które same muszą skonstruować! Wykluczamy naiwne szukanie selektora lub rodziny selektywnej przez sprawdzanie kolejnych rodzin i proste próby derandomizacji wspomnianych konstrukcji probabilistycznych, gdyż używały dużej liczby bitów losowych.

Przedstawimy konstrukcję explicite (n, k) -selektorów, pochodzącą z [Ind02]. Z Faktu 2 daje ona również konstrukcję rodzin (n, k) -selektywnych. Rozmiar konstruowanych selektorów wynosi $\mathcal{O}(k \log^3 n)$, więc rozmiar uzyskanych rodzin selektywnych jest tego samego rzędu. Prezentowana konstrukcja w pewnym sensie polega na derandomizacji losowej konstrukcji z Twierdzenia 5, jednak jest wysoce nietrywialna. Korzystamy z

- *p-kolidujących rodzin funkcji*, tj. pewnych rodzin funkcji haszujących, zachowujących się z grubsza jak funkcje losowe, oraz
- *rozpraszaczy (dispersers)*, tj. grafów dwudzielnych o stałym stopniu i dużym współczynniku ekspansji, pomocnych w wydajnej derandomizacji.

Definicje obu obiektów prezentujemy poniżej. W tym rozdziale, w przeciwieństwie do poprzednich, dla wygody elementy zbiorów numerujemy od zera — będziemy się zajmować m.in. resztami z dzielenia. W szczególności przyjmujemy teraz $[n] = \{0, 1, \dots, n-1\}$.

Definicja 7. (n, m, d, k, ϵ) -rozpraszacz to nieskierowany graf dwudzielny $G = (A, B, E)$ taki, że

- (1) $|A| = n$, $|B| = m$
- (2) $\forall Z \subseteq A, |Z|=k |\Gamma_G(Z)| \geq (1-\epsilon)m$
- (3) $\forall v \in A \deg(v) = d$

Używać będziemy tylko $(n, m, d, \frac{m}{2}, \frac{3}{4})$ -rozpraszaczy, poniżej przedstawiamy ich uproszczoną definicję.

Definicja 8. (n, m, d) -rozpraszacz, to nieskierowany graf dwudzielny $G = (A, B, E)$ taki, że

- (1) $|A| = n$, $|B| = m$

- (2) $\forall Z \subseteq A, |Z|=m/2 |\Gamma_G(Z)| \geq \frac{m}{4}$
 (3) $\forall v \in A \deg(v) = d$

Definicja 9. Rodzinę $G = \{g_0, g_1, \dots, g_{r-1}\}$ funkcji $g_j: A \rightarrow [u]$ nazwiemy p -kolidującą rodziną funkcji o parametrach (r, u) , jeśli przy losowaniu funkcji g z rodziny G z rozkładem jednostajnym zachodzi

$$\forall x, y \in A, x \neq y \Pr_{g \in G} [g(x) = g(y)] \leq p.$$

Fakt 10. Dla dowolnego $p \in (0, 1]$ istnieje p -kolidująca rodzina funkcji o parametrach (r, u) , gdzie $r = \mathcal{O}\left(\frac{\log n}{p}\right)$, $u = \mathcal{O}\left(\frac{\log^2 n}{p}\right)$, o ile n , moc wspólnej dziedziny funkcji z tej rodziny, jest dostatecznie duże.

Fakt ten uzupełniamy o dowód, pominięty w [Ind02].

Dowód. Weźmy zbiór $P = \{q_0, q_1, \dots, q_{r-1}\}$ liczb pierwszych, taki że $|P| = r = \Theta\left(\frac{\log n}{p}\right)$ oraz dla każdego $q_i \in P$, $q_i = \Theta\left(\frac{\log^2 n}{p}\right)$. Następnie zdefiniujemy g_j jako $g_j(x) = x \bmod q_j$. Ustalmy teraz niektóre stałe, ukryte w notacji Θ : niech $q_i \geq \alpha \cdot \frac{\log^2 n}{p}$ oraz $r \geq \beta \cdot \frac{\log n}{p}$. Weźmy dowolne $x, y \in A$, $x \neq y$ i dla ustalenia uwagi niech $x > y$. Oczywiście zachodzi $0 < x - y < n$.

Wystarczy, że pokażemy, że przy losowym wyborze $q_i \in P$ z rozkładem jednostajnym, $\Pr_{q_i \in P} [q_i | x - y] \leq p$. Skoro $0 < x - y < n$ a wszystkie liczby pierwsze z P są nie mniejsze niż $\alpha \frac{\log^2 n}{p}$, to dzielnikami $x - y$ może być najwyżej

$$\log_{\alpha \cdot \frac{\log^2 n}{p}} n = \frac{\log n}{\log \alpha + 2 \log \log n + \log \frac{1}{p}}$$

spośród nich. Zatem

$$\begin{aligned} \Pr_{q_i \in P} [q_i | x - y] &\leq \frac{1}{|P|} \cdot \frac{\log n}{\log \alpha + 2 \log \log n + \log \frac{1}{p}} \leq \\ &\leq \frac{p}{\beta \log n} \cdot \frac{\log n}{\log \alpha + 2 \log \log n + \log \frac{1}{p}} = \frac{p}{\beta \left(\log \alpha + 2 \log \log n + \log \frac{1}{p} \right)} \leq p. \end{aligned}$$

Wykażemy jeszcze, że faktycznie istnieje $\Omega\left(\frac{\log n}{p}\right)$ liczb pierwszych rzędu $\Theta\left(\frac{\log^2 n}{p}\right)$.

Przyjmijmy więc, że każda z nich jest nie mniejsza niż $\frac{\alpha}{e} \cdot \frac{\log^2 n}{p}$. Wtedy liczb pierwszych w interesującym nas przedziale jest

$$\begin{aligned} &\pi\left(\alpha \frac{\log^2 n}{p}\right) - \pi\left(\frac{\alpha}{e} \cdot \frac{\log^2 n}{p}\right) \sim \\ &\sim \alpha \cdot \frac{\log^2 n}{p} \cdot \frac{1}{\ln \alpha + 2 \ln \log n + \ln \frac{1}{p}} - \frac{\alpha}{e} \cdot \frac{\log^2 n}{p} \cdot \frac{1}{\ln \alpha + 2 \ln \log n + \ln \frac{1}{p} - 1} \approx \\ &\approx \alpha \left(1 - \frac{1}{e}\right) \cdot \frac{\log^2 n}{p} \cdot \frac{1}{\ln \alpha + 2 \ln \log n + \ln \frac{1}{p}} = \omega\left(\frac{\log n}{p}\right). \end{aligned}$$

□

7.2. Właściwa konstrukcja.

Twierdzenie 23. Mając dany (n, k, d) -rozpraszacz, można w czasie wielomianowym skonstruować (n, k) -selektor rozmiaru $\mathcal{O}(kd^3 \log^3 n)$.

Dowód. Niech (A, B, E) będzie dowolnym (n, k, d) -rozpraszaczem. Dla każdego $x \in A$ określamy porządek na jego d sąsiadach. Określamy funkcje $h_0, h_1, \dots, h_{d-1}: A \rightarrow B$ tak, że $h_i(x)$ jest i -tym sąsiadem x . Niech wreszcie $G = \{g_0, g_1, \dots, g_{r-1}\}$ będzie p -kolidującą rodziną funkcji $g_j: A \rightarrow [u]$, dla $p = \frac{1}{48d}$. Dzięki Faktowi 10 możemy przyjąć, że $r = \mathcal{O}(d \log n)$, $u = \mathcal{O}(d \log^2 n)$.

Selektor konstruujemy następująco. Niech $f_{i,j}: A \rightarrow [k] \times [u]$ dla $i \in [d]$, $j \in [r]$ będzie funkcją określoną jako

$$f_{i,j}(x) = (h_i(x), g_j(x)) .$$

Selektor \mathcal{S} zawiera wszystkie zbiory postaci

$$S_{s,t,i,j} = f_{i,j}^{-1}(s, t)$$

dla $i \in [d]$, $j \in [r]$ $s \in [\frac{k}{2}]$, $t \in [u]$.

Oczywiste jest, że rozmiar \mathcal{S} wynosi $\mathcal{O}(kd^3 \log^3 n)$. Pokażemy, że jest (n, k) -selektorem. W tym celu ustalmy zbiory X, Y , jak w Definicji 3. Zauważmy, że z własności ekspansji rozpraszacza, tj. własności (2) w Definicji 8, wynika, że dla dowolnego $Z \subseteq A$, $|Z| \geq \frac{k}{2}$, zachodzi $|\Gamma_G(Z)| \geq \frac{k}{4}$, więc dla jednej spośród d funkcji h_i zachodzi $|h_i(Z)| \geq \frac{k}{4d}$.

Weźmy tę funkcję dla $Z = X$ i zdefiniujmy $X_b = h_i^{-1}(b) \cap X$ oraz $Y_b = h_i^{-1}(b) \cap Y$, dla $b \in B$. Słownie, X_b to zbiór tych elementów ze zbioru X , które h_i przeprowadza na b i analogicznie Y_b to zbiór tych elementów ze zbioru Y , które h_i przeprowadza na b . Określimy B' jako zbiór tych $b \in B$, na które h_i przeprowadza co najmniej jeden element zbioru X , tj. $B' = \{b \in B: X_b \neq \emptyset\}$. Zauważmy, że $|B'| \geq \frac{k}{4}$, bo $|h_i(X)| \geq \frac{k}{4}$.

Ponieważ zbiory X_b są rozłączne, średnia moc X_b dla $b \in B'$, tj. średnia moc niepustych zbiorów X_b , wynosi $\frac{|h_i^{-1}(B) \cap X|}{|B'|} \leq \frac{|X|}{|B'|} \leq \frac{k}{|B'|} \leq 4d$. Stąd zbiorów X_b o mocy nie mniejszej niż $12d$ może być najwyżej $\frac{1}{3}|B'|$, zatem co najmniej $\frac{2}{3}|B'|$ spośród niepustych zbiorów X_b ma moce mniejsze niż $12d$. To samo rozumowanie wykazuje, że zbiorów Y_b dla $b \in B'$ o mocach mniejszych niż $12d$ jest również co najmniej $\frac{2}{3}|B'|$. W takim razie dla co najmniej $\frac{1}{3}|B'|$ elementów $b \in B'$ zachodzi $|X_b| < 12d$ i $|Y_b| < 12d$. Oznaczmy je przez B'' , tj. niech

$$B'' := \{b \in B': |X_b| < 12d \wedge |Y_b| < 12d\} .$$

Dla każdego $x \in B''$ zdefiniujmy indykator $I_x \in \{0, 1\}$, zależny od losowego wyboru z rozkładem jednostajnym $j \in [r]$. Określamy $I_x = 1$ wtedy i tylko wtedy, gdy

$$(8) \quad g_j(x) \notin g_j(X_{h_i(x)} \cup Y_{h_i(x)} \setminus \{x\}) .$$

Pokażemy, że gdy (8) zachodzi dla pary (x, j) , to x jest jedynym elementem z zbioru $X \cup Y$ spełniającym równość $f_{i,j}(z) = f_{i,j}(x)$, czyli że $S_{h_i(x), g_j(x), i, j} = f_{i,j}^{-1}(h_i(x), g_j(x))$ wybiera X i omija Y . Istotnie, jeśli zachodzi (8) i dla $z \in X \cup Y$, $z \neq x$, zachodzi $h_i(x) = h_i(z)$, to $g_j(x) \neq g_j(z)$, skąd $f_{i,j}(x) \notin f_{i,j}(X \cup Y \setminus \{x\})$.

Z własności rodziny G wynika, że

$$\Pr_{g \in G} [g(x) \in g(X_{h_i(x)} \cup Y_{h_i(x)} \setminus \{x\})] \leq 24d \cdot p = \frac{1}{2} ,$$

czyli dla pewnego $x \in B''$ funkcja $g \in G$ spełniająca (8) istnieje, bo

$$\mathbb{E}_{g \in G} \left[\sum_{x \in B''} I_x \right] \geq \frac{|B''|}{2} > 0 .$$

□

Twierdzenie 23 w istocie gwarantuje, że można skonstruować (n, k) -selektor w czasie wielomianowym, gdyż wielomianowe konstrukcje rozpraszaczy o stałym stopniu są znane od dawna. Jak zaznaczono w [Ind02], zastosowanie ekstraktorów bądź silnych ekstraktorów, tj. rozpraszaczy o dodatkowych, silniejszych własnościach, pozwala uzyskać nieco mniejszy rozmiar konstruowanych selektorów. Same konstrukcje rozpraszaczy i ekstraktorów są nawet wydajniejsze niż zakładamy — zwykle mają znajdować zastosowanie w sytuacji, w której rozmiar grafu jest wykładniczo duży w stosunku do rozmiaru instancji problemu. Dlatego w czasie polilogarytmicznym względem rozmiaru grafu można dla dowolnego wierzchołka grafu v i dowolnej liczby i obliczyć identyfikator i -tego sąsiada wierzchołka v .

Więcej informacji na temat rozpraszaczy i ekstraktorów, wraz z bogatą listą odnośników, można znaleźć w przeglądowych artykułach na temat ekstraktorów: [NT99] i nowszym [Sha04]. Z kolei w [CK05] podano konstrukcję uogólnionych selektorów o mniejszym rozmiarze niż z cytowanej przez nas konstrukcji z [Ind02].

8. GRAFY NIESKIEROWANE

W tym rozdziale przedstawimy kilka wyników dotyczących grafów nieskierowanych. Okazuje się, że rozgłaszanie można w nich przeprowadzić wydajniej niż w grafach skierowanych. Ciekawe jest również, że długo nie było znane żadne nietrywialne ograniczenie dolne na czas powiadomienia dla grafów nieskierowanych.

Ponownie przyjmujemy konwencję z podrozdziału o modelu z detekcją kolizji, że μ oznacza dowolny sygnał. Jednak teraz detekcja kolizji nie jest dostępna, za to wszystkie prezentowane w tym rozdziale protokoły opierają się na przekazywaniu żetonu i tym sposobem unikają kolizji. Dlatego μ zawsze oznacza sygnał, który jest dowolny, tj. ważne jest tylko czy jest nadawany.

8.1. Liniowy czas przy spontanicznej komunikacji. Wszystkie dotychczasowe protokoły nie korzystały ze spontanicznej komunikacji, tj. węzły nie nadawały w nich żadnych komunikatów, dopóki nie poznały wiadomości m . Teraz omówimy protokół EXPLORE-AND-EXPAND z [CGGPR00], który gwarantuje liniowy czas powiadomienia i terminacji w grafach nieskierowanych dzięki spontanicznej komunikacji.

Protokół EXPLORE-AND-EXPAND bazuje na rozproszonym przeszukiwaniu grafu w głąb [Awe85]. Łatwo o nim myśleć gdy n jest znane — wtedy w ciągu n rund wykonuje on jedną fazę ROUND-ROBIN, w której każdy węzeł nadaje swój identyfikator. Po ukończeniu tej fazy, każdy węzeł zna identyfikatory swoich sąsiadów, a ta wiedza wystarcza do przeprowadzenia w czasie $\mathcal{O}(n)$ rozgłaszania wiadomości m poprzez rozproszone wyszukiwanie w głąb. Teraz opiszemy protokół szczegółowo, dla nieznanego n .

Jak zwykle, wykonujemy kolejne fazy, gdzie faza i wystarcza dla grafów o 2^i wierzchołkach. Przez G_i oznaczać będziemy spójną składową źródła s w podgrafie grafu G indukowanym na wierzchołkach o identyfikatorach mniejszych nie większych niż 2^i . Spełniony będzie następujący niezmiennik:

Niezmiennik 2. *Po skończonej fazie k*

- *skonstruowane jest T_k , drzewo rozpinające grafu G_k oraz cykl Eulera C_k zaczynający i kończący się w źródle, powstały przez podwojenie krawędzi T_k*
- *każdy węzeł cyklu C_k otrzymał i zna etykietę będącą jego numerem w cyklu, gdzie numerację zaczynamy od s*
- *każdy węzeł z G_k zna identyfikatory swoich sąsiadów w G_k*
- *każdy węzeł z G_k zna wiadomość m*

Teraz opiszemy fazę k Protokołu EXPLORE-AND-EXPAND przy założeniu, że niezmiennik jest spełniony dla $k - 1$. Faza podzielona jest na trzy tury:

- Tura A składa się z 2^{k-1} rund, numerowanych od $2^{k-1}+1$ do 2^k . W rundzie j węzeł j nadaje swój identyfikator. Po turze A każdy węzeł z G_k zna identyfikatory swoich sąsiadów w G_k .
- Tura B składa się z 2^k rund, numerowanych od 1 do 2^k . Węzeł z cyklu C_{k-1} nazwiemy aktywnym w rundzie j fazy k , jeśli odebrał jakikolwiek sygnał (czyli μ bądź identyfikator) od początku tej fazy, tj. w turze A lub w $j-1$ początkowych rundach tury B. Każdy węzeł aktywny ma i zna swoją etykietę z cyklu C_{k-1} . Węzeł aktywny o etykietce j nadaje μ w rundzie j tury B. Ponieważ etykiety opisują porządek w cyklu C_{k-1} a źródło ma (między innymi) maksymalną etykietę, 2^k rund wystarcza, by zostało uaktywnione.
- Tura C następuje, jeśli źródło zostało uaktywnione w turze B. Jeśli tak jest, m jest przekazywane w G_k w oparciu o rozproszone przeszukiwanie w głąb — każdy węzeł $v \in V(G_k)$ ma listę sąsiadów Q_v , do których chce przekazać m . Lista ta może być dowolnie uporządkowana i początkowo dla węzła v zawiera wszystkich sąsiadów v z G_k . Gdy węzeł odbierze od swojego sąsiada informację, że ten już został odwiedzony, wykreśla go z listy. W grafie krąży żeton (z licznikiem) uprawniający do transmisji, początkowo posiadany przez źródło. Stan licznika wyznacza etykiety węzłów, przy czym jeden węzeł może mieć kilka etykiet. Wszystkich etykiet jest o 1 więcej niż wynosi długość cyklu C_k , czyli najwyżej $2 \cdot 2^k$. Źródło ma (m.in.) etykietę 1. Węzeł v , gdy otrzyma wiadomość m oraz żeton z licznikiem c , postępuje następująco:
 - wykreśla u , węzeł od którego otrzymał żeton z Q_v (tak samo postępuje każdy sąsiad u , bo u został odwiedzony)
 - przypisuje sobie etykietę $c+1$
 - nadaje sygnał $\langle m, v, c+1, w \rangle$, gdzie w jest pierwszym węzłem z Q_v , lub $w = u$, jeśli Q_v jest pusta.
 - gdy żeton wraca do źródła a Q_s jest pusta, tura C się kończy

Nadawany przez v sygnał $\langle m, v, c+1, w \rangle$ pełni dwie funkcje:

- przekazuje m wszystkim sąsiadom v i informuje ich, że v został odwiedzony
- oraz przekazuje żeton i stan licznika do węzła w .

Niezmiennik pozostaje spełniony w oczywisty sposób. Wszystkie węzły zostaną poinformowane w $\lceil \log n \rceil$ -tej fazie. W kolejnych fazach żaden węzeł nie nada sygnału w turze A, przez co żaden nie uczyni tego również w turze B. To oznacza, że tura C w ogóle się nie rozpocznie. Warto zauważyć, że źródło i pozostałe węzły będą mimo to w nieskończoność oczekiwać na sygnał od potencjalnie istniejących kolejnych węzłów. Sytuacji takiej nie da się uniknąć — więcej na ten temat w dalszym rozdziale.

Poprawność została już omówiona. Każda z tur fazy k trwa $\mathcal{O}(2^k)$, więc otrzymujemy twierdzenie:

Twierdzenie 24. *Protokół EXPLORE-AND-EXPAND przeprowadza rozgłaszanie i terminuje w czasie $\mathcal{O}(n)$ dla dowolnego grafu nieskierowanego G o n wierzchołkach, korzystając ze spontanicznej komunikacji.*

8.2. Symulowanie detekcji kolizji. W [KP02] wprowadzono (dla grafów nieskierowanych) technikę, zwaną „symulacją detekcji kolizji”. Nazwa bierze się stąd, że technika ta pozwala węzłowi sieci poznać liczbę własnych sąsiadów, o ile zna identyfikator jednego z nich — mniej więcej to umożliwia detekcja kolizji, gdy wszyscy sąsiedzi jednocześnie nadają, nie wymagając przy tym znajomości jednego z sąsiadów. Warunek ten nie jest z resztą kłopotliwy, gdyż każdy węzeł, który otrzymał

```

Dane niejawne:  $G(V, E)$ , nieskierowany
Dane jawne:  $n, s, m$ 

 $T := \{s\};$ 
 $C := \{s\};$ 

/* Źródło  $s$  ma etykietę 1 */
/* Przez  $label(j)$  oznaczamy węzeł, który ma obecnie etykietę  $j$  */
for  $k = 1, 2, \dots$  do /* kolejne fazy */

    for  $v = 2^{k-1} + 1, 2^{k-1} + 2, \dots, 2^k$  do /* tura A */

         $v$  nadaje swój identyfikator, czyli  $v$ ;

    for  $j = 1, 2, \dots, 2^k$  do /* tura B */

        if  $label(j)$  odebrał w tej fazie cokolwiek then
             $label(j)$  nadaje  $\mu$ ;

        if  $s$  odebrał w tej fazie cokolwiek then /* następuje tura C */

             $T := \emptyset;$ 
             $C := \emptyset;$ 

            foreach  $v \in V(G_k)$  do
                 $v$  tworzy  $Q_v$ , kolejkę swoich sąsiadów z  $G_k$  w dowolnym porządku;
            /* Źródło  $s$  otrzymuje żeton  $\langle m, \perp, 0, s \rangle$  */
            while Żeton nie jest w posiadaniu źródła  $s$  lub  $Q_s$  jest niepusta do

                /* Żeton  $\langle m, u, c, v \rangle$  jest w posiadaniu węzła  $v$  */
                 $v$  przypisuje sobie etykietę  $c + 1$ ;

                if  $Q_v$  jest niepusta then
                     $w := \text{First}(Q_v);$ 
                else
                     $w := u;$ 

                 $v$  nadaje sygnał  $\langle m, v, c + 1, w \rangle$ ;

                /* Sygnał ten odbierają sąsiedzi  $v$  */
                Sąsiedzi  $v$  usuwają  $v$  ze swoich kolejek;
                 $C := C \cup \{(v, w)\};$ 

                if  $w \neq u$  then
                     $T := T \cup \{(v, w)\};$ 

             $label(c + 1) := s$ , gdzie  $c$  jest etykietą, którą  $s$  odebrał ostatnio;

```

Protokół 10: Protokół EXPLORE-AND-EXPAND

wiadomość m może wraz z nią otrzymać identyfikator sąsiada, który przesłał mu m . Tylko źródło, które nie otrzymuje m od nikogo, nie może skorzystać z tej wiedzy.

W dotychczasowych protokołach wystarczało, by źródło w pierwszej rundzie nadało m i następnie się zdezaktywowało, jednak teraz postępować będziemy inaczej! Będziemy, podobnie jak w Protokole EXPLORE-AND-EXPAND z poprzedniego podrozdziału, wykonywać rozproszone wyszukiwanie w głąb, z użyciem przekazywania żetonu. To oznacza, że źródło musi wiedzieć, do kogo adresuje każdą wiadomość

i że będzie ich wysyłać wiele. Krótko mówiąc, źródło musi najpierw w jakiś sposób poznać identyfikator jednego ze swych sąsiadów. Poniżej prezentujemy sposób w jaki może to zrobić — Protokół ANCHOR.

W pierwszej rundzie źródło wysyła ustalony sygnał, który nakazuje węzłowi o identyfikatorze i nadać i w rundzie $2i$. Gdy źródło odbierze tę wiadomość dla najmniejszego i będącego jego sąsiadem, w rundzie $2i + 1$ nadaje kolejny ustalony sygnał, nakazujący przerwanie nadawania identyfikatorów. W oczywisty sposób nie dochodzi do kolizji, a źródło w czasie $\mathcal{O}(n)$ poznaje identyfikator jednego ze swoich sąsiadów.

Dane niejawne: $G(V, E)$, nieskierowany

Dane jawne: s

Wynik: s poznaje najmniejszy z identyfikatorów swoich sąsiadów; jeśli s jest jedynym węzłem, protokół terminuje przez zawieszenie

```

/* runda 1 */
s nadaje sygnał, który nakazuje węzłowi  $i$  nadać sygnał w rundzie  $2i$ ;
if  $i$  odebrał powyższy sygnał then
     $v$  nadaje  $v$  w rundzie  $2i$ ;
if  $s$  odbiera po raz pierwszy sygnał w rundzie  $2i^*$  then /* w rundzie  $2i^* + 1$ 
    (kolejnej)  $s$  nakazuje przerwanie protokołu */
     $s$  nadaje sygnał nakazujący sąsiadom przerwanie protokołu;
return  $i^*$ ;
/* Jeśli  $s$  jest jedynym węzłem, w nieskończoność czeka na sygnał.
Ponieważ źródło od tej pory nie nadaje sygnałów, zgodnie
z definicją protokołu dokonał rozgłaszania i terminował. */

```

Protokół 11: Protokół ANCHOR

Gdy węzeł v zna swojego sąsiada w , by poznać liczbę swych pozostałych sąsiadów wykonuje razem z nimi i z w 3-rundowy Protokół ECHO(v, w). W pierwszej rundzie v nadaje sygnał rozpoczęcia protokołu. Słyszą go sąsiedzi v i reagują następująco: w drugiej rundzie każdy z nich z wyjątkiem w nadaje swój identyfikator a w trzeciej rundzie każdy z nich, w tym w , nadaje swój identyfikator. v na podstawie tego co usłyszał w tych dwóch rundach wnioskuje o liczbie swoich sąsiadów:

- jeśli v słyszy (u, \perp) dla pewnego u , wie, że u jest jego jedynym sąsiadem oprócz w
- jeśli v słyszy (\perp, w) , wie, że w jest jego jedynym sąsiadem
- jeśli v słyszy (\perp, \perp) , wie, że ma co najmniej dwóch sąsiadów oprócz w .

Można ograniczyć przedział, w którym badamy liczbę sąsiadów węzła v . Wystarczy, by w pierwszej rundzie v w komunikacie nadał krańce tego przedziału, czyli parę (x, y) . W protokole wezmą udział tylko jego sąsiedzi o identyfikatorach z $[x, y]$ oraz w — w postępuje tak samo niezależnie od tego, czy znajduje się w przedziale $[x, y]$ czy nie. Pseudokod prezentujemy dla wariantu z przedziałem.

W [KP03a] ci sami autorzy wykorzystali tę technikę do rozgłaszania w sieciach symetrycznych bez spontanicznej komunikacji, uzyskując czas powiadomienia $\mathcal{O}(n \log n)$. Symulację detekcji kolizji wykorzystali do selekcji jednego z sąsiadów poprzez wyszukiwanie binarne. Wyszukiwanie binarne przeprowadza się za pomocą Protokołu BINARY-SELECTION, którego szczegóły prezentujemy w pseudokodzie.

```

Dane niejawne:  $G(V, E)$ , nieskierowany
Dane jawne: węzeł  $v$  i jego sąsiad  $w$ , liczby  $x$  i  $y$ 
Wynik:  $v$  poznaje liczbę swoich sąsiadów o identyfikatorach z przedziału
            $[x, y]$ , różnych od  $w$ : wie, czy jest ich 0, 1, czy co najmniej 2; gdy  $v$ 
           ma dokładnie jednego takiego sąsiada, dodatkowo poznaje jego
           identyfikator

/* runda 1 */
v nadaje  $[x, y]$ ;
/* runda 2 */
foreach  $z \in N(v) \cap [x, y] \setminus \{w\}$  do
  z nadaje swoją etykietę, czyli  $z$ ;
/* runda 3 */
foreach  $z \in N(v) \cap [x, y] \cup \{w\}$  do
  z nadaje swoją etykietę, czyli  $z$ ;
/*  $v$  wnioskuje o liczbie sąsiadów */
switch (sygnały odebrane przez  $v$  w rundach 2 i 3 odpowiednio) do
  case  $(u, \perp)$  /*  $u$  jest jedynym sąsiadem  $v$  w  $[x, y]$  różnym od  $w$  */
    return  $u$ ;
  case  $(\perp, w)$  /*  $v$  nie ma sąsiadów w  $[x, y]$  różnych od  $w$  */
    return  $\perp$ ;
  case  $(\perp, \perp)$  /*  $v$  ma co najmniej dwóch sąsiadów w  $[x, y]$  różnych
    od  $w$  */
    return  $?$ ;

```

Protokół 12: Protokół ECHO

Używając go, znając identyfikator swego sąsiada w węzeł v może poznać identyfikator jednego z pozostałych swoich sąsiadów przy założeniu, że wszystkie identyfikatory nie przekraczają n . W logarytmicznej względem n liczbie kroków dowolny węzeł, znając identyfikator jednego ze swych sąsiadów, może w ten sposób poznać identyfikator innego. Co więcej, przez wyłączenie z protokołu tych sąsiadów, których identyfikatory już zna, może poznać identyfikatory wszystkich swoich sąsiadów o identyfikatorach nie większych niż n . Na poznanie każdego z nich poświęca czas $\mathcal{O}(\log n)$.

Dysponując Protokołem BINARY-SELECTION, można symulować Protokół EXPLORE-AND-EXPAND w czasie $\mathcal{O}(n \log n)$ bez użycia spontanicznej komunikacji. Do tego symulację można przeprowadzić w jednej fazie, tj. bez powtórzeń dla $n = 1, 2, 4, 8, \dots$. W symulacji rezygnujemy z tur A i B. Tura A służyła węzłom z G_k do poznania identyfikatorów swoich sąsiadów. Teraz węzły poznają identyfikatory sąsiadów za pomocą Protokołu BINARY-SELECTION, gdy ich potrzebują. Z kolei tura B służyła poinformowaniu źródła, czy należy wykonać turę C dla zwiększonego n . Można zrezygnować i z niej, skoro wystarcza jedno wykonanie tury C. Symulację, czyli Protokół SELECT-AND-SEND prezentujemy poniżej w sposób zbliżony do Protokołu EXPLORE-AND-EXPAND:

- stosując Protokół ANCHOR, źródło poznaje j , minimalny identyfikator ze zbioru swoich sąsiadów w czasie $\mathcal{O}(j)$, czyli z pewnością $\mathcal{O}(n)$

```

Dane niejawne:  $G(V, E)$ , nieskierowany
Dane jawne: węzeł  $v$  i jego sąsiad  $w$ , ograniczenie górne na przedział,
                w którym szukamy sąsiadów:  $l$ 
Wynik:  $v$  poznaje identyfikator swojego sąsiada o identyfikatorze nie
                większym niż  $l$  i różnego od  $w$ , lub dowiadyuje się, że nie ma takiego
                sąsiada

 $b := \text{ECHO}(v, w, 1, l)$ ;
if  $b = \perp$  then return  $\perp$ ; /*  $v$  nie ma sąsiadów różnych od  $w$ 
                o identyfikatorach nie większych niż  $l$  */
 $x := 1$ ;
 $y := \lceil l/2 \rceil$ ;
while  $b \in \{?, \perp\}$  do
     $b := \text{ECHO}(v, w, x, y)$ ;
    switch  $b$  do
        case  $\perp$  /* Sąsiedzi  $v$  mają identyfikatory większe niż  $y$  */
             $(x, y) := (y + 1, y + \lceil \frac{y-x+1}{2} \rceil)$ ;
        case  $?$  /*  $v$  ma więcej niż jednego sąsiada różnego od  $w$ 
                o identyfikatorze z  $[x, y]$  */
             $(x, y) := (x, \lceil \frac{y+x-1}{2} \rceil)$ ;
    /* Pętla zakończyła się, bo wynikiem Echo było  $u$  */
return  $b$ ;

```

Protokół 13: Protokół BINARY-SELECTION

- wykonujemy rozproszone przeszukanie grafu w głąb z użyciem przekazywania żetonu (jak w EXPLORE-AND-EXPAND), przy czym stosujemy następujące modyfikacje:
 - węzeł v , gdy otrzyma żeton po raz pierwszy, oraz źródło s , po tym gdy pozna j , wyznacza minimalne takie k , że wszyscy jego sąsiedzi mają identyfikatory nie większe niż 2^k . W tym celu wykonuje $\text{ECHO}(v, \text{parent}(v), 2^k + 1, \infty)$, dla $k = 0, 1, 2, \dots$, aż stwierdzi, że $|N(v) \cap [2^{k+1} + 1, \infty) \setminus \{\text{parent}(v)\}| = 0$, gdzie $\text{parent}(v)$ to węzeł od którego $v \neq s$ otrzymał żeton, lub j dla s . Każdemu z węzłów ten krok zajmuje $\mathcal{O}(\log n)$ rund. Węzły, które miały już żeton, nie nadają, oczywiście z wyjątkiem $\text{parent}(v)$.
 - węzeł v , który zna już ograniczenie górne na identyfikatory swoich sąsiadów, wyznacza kolejno ich identyfikatory, gdy ma przekazać żeton, przez wywołanie Protokołu BINARY-SELECTION. Również tu węzły, które już otrzymały żeton, nie nadają. Wyznaczenie każdego identyfikatora zajmuje $\mathcal{O}(n \log n)$ kroków. Ponieważ węzły, które już otrzymały żeton ignorują wywołania ECHO), sumaryczny czas wynosi $\mathcal{O}(n \log n)$.

Terminacja tego protokołu jest oczywista. W [KP03a] był on prezentowany przy założeniu, że n znane jest wszystkim węzłom. My wprowadziliśmy drobną modyfikację, która pozwala zrezygnować ze znajomości n .

Twierdzenie 25. *Protokół SELECT-AND-SEND przeprowadza rozgłaszanie bez spontanicznej komunikacji i terminuje w czasie $\mathcal{O}(n \log n)$ dla dowolnego grafu nieskierowanego G o n wierzchołkach.*

```

Dane niejawne:  $G(V, E)$ , symetryczny
Dane jawne:  $s, m$ 

/* węzły, które już miały żeton, ignorują wywołania ECHO i
  BINARY-SELECTION */
s* := Anchor();
/* s poznał s*, najmniejszy z identyfikatorów swoich sąsiadów;
  jeśli s nie ma sąsiadów, protokół trywialnie dokonał
  rozgłaszania i terminuje przez zawieszenie */
l_s := 1;
repeat
  l_s := 2l_s;
  b := ECHO(s, s*, l_s + 1, ∞);
until b = ⊥ ;
/* sąsiedzi s mają identyfikatory nie większe niż l_s */
s nadaje sygnał < m, s, s* >;
while Żeton nie jest w posiadaniu źródła s lub nie miał go jeszcze jeden z
sąsiadów s do
  /* Żeton < m, u, v > jest w posiadaniu węzła v */
  if v posiada żeton po raz pierwszy then
    l_v := 1;
    repeat
      l_v := 2l_v;
      b := ECHO(v, u, l_v + 1, ∞);
    until b = ⊥ ;
  /* sąsiedzi v, którzy nie mieli jeszcze żetonu mają
    identyfikatory nie większe niż l_v */
  w := BINARY-SELECTION(v, u, l_v);
  if w ≠ ⊥ then
    v nadaje sygnał < m, v, w >;
  else
    v nadaje sygnał < m, v, u >;

```

Protokół 14: Protokół SELECT-AND-SEND

8.3. Pełne nieskierowane grafy warstwowe oraz ograniczenia dolne. W [KP03a] opisaną wcześniej technikę zastosowano również do pełnych nieskierowanych grafów warstwowych, uzyskując Protokół UCLN o czasie powiadomienia $\mathcal{O}(n + D \log n)$. Protokół UCLN jest prostą modyfikacją Protokołu SELECT-AND-SEND — korzysta się z tego, że wystarczy by w każdej warstwie żeton otrzymał jeden węzeł. W jednej rundzie przekaże on m wszystkim węzłom z następnej warstwy a potem, przy użyciu Protokołu BINARY-SELECTION przekaże żeton temu spośród nich, który ma najmniejszy identyfikator. Przekazanie żetonu z warstwy L_i do L_{i+1} zabiera $\mathcal{O}(\log n)$ rund dla $0 < i < D$, zaś źródło potrzebuje czasu $\mathcal{O}(n)$ by poznać j . W sumie czas (powiadomienia i terminacji) wynosi $\mathcal{O}(n + D \log n)$.

Co więcej, węzeł v nie musi znać ograniczenia górnego na identyfikatory swoich sąsiadów — wystarczy by wyznaczył najmniejszy z nich, tj. poprzestał na przedziale identyfikatorów $[2^k]$ dla minimalnego k takiego, że zbiór jego sąsiadów z kolejnej warstwy o identyfikatorach z $[2^k]$ jest niepusty. By zagwarantować terminację, v musi tylko wcześniej stwierdzić, że w ogóle posiada sąsiadów w kolejnej warstwie. Również to osiąga za pomocą Procedury ECHO, tyle że bez ograniczeń na identyfikatory sąsiadów. Jeśli nie ma sąsiadów w kolejnej warstwie, to znaczy że v jest w L_D , tj. warstwie ostatniej. Wtedy protokół jest zakończony.

Twierdzenie 26. *Protokół UCLN przeprowadza rozgłaszanie bez spontanicznej komunikacji i terminuje w czasie $\mathcal{O}(n + D \log n)$ dla dowolnego pełnego nieskierowanego grafu G o n wierzchołkach i maksymalnej odległości od źródła D .*

```

Dane niejawne:  $G(V, E)$ , nieskierowany
Dane jawne:  $s, m$ 

/* Węzły, które odebrały  $m$ , ignorują wywołania ECHO i
   BINARY-SELECTION */
s* :=ANCHOR();
/*  $s$  poznał  $s^*$ , najmniejszy z identyfikatorów swoich sąsiadów;
   jeśli  $s$  nie ma sąsiadów, protokół trywialnie dokonał
   rozgłaszania i terminuje przez zawieszenie */
s nadaje sygnał  $\langle m, s, s^* \rangle$ ;
v := s;
u := s*;
while ECHO( $v, u, 1, \infty$ )  $\neq \perp$  do
  /* Żeton  $\langle m, u, v \rangle$  posiada węzeł  $v$  */
   $l_v := 1$ ;
  repeat
     $l_v := 2l_v$ ;
     $b := \text{ECHO}(v, u, 1, l_v)$ ;
  until  $b \neq \perp$ ;
  /* najmniejszy identyfikator sąsiada  $v$  z kolejnej warstwy jest
     nie większy niż  $l_v$  */
   $w := \text{BINARY-SELECTION}(v, u, l_v)$ ;
  v nadaje sygnał  $\langle m, v, w \rangle$ ;

```

Protokół 15: Protokół UCLN

Zauważmy, że czas $\mathcal{O}(n + D \log n)$ jest mniejszy niż dolne ograniczenie $\Omega(n \log D)$ z twierdzenia 14 dla $D = o(n)$. Twierdzenie 14 dotyczyło pełnych grafów warstwowych, ale skierowanych. W [CMS01] stawiano hipotezę, że ograniczenie to pozostaje w mocy również dla nieskierowanych pełnych grafów warstwowych. W [KP03a] podano za to dolne ograniczenie na czas rozgłaszania bez spontanicznej komunikacji w grafach nieskierowanych. Twierdzenie to cytujemy bez dowodu. Podkreślamy tylko, że konstruowane w dowodzie sieci nie są pełnymi nieskierowanymi grafami warstwowymi.

Twierdzenie 27. *Dla dowolnego protokołu rozgłaszania P bez spontanicznej komunikacji oraz dowolnych n i D takich, że $D \geq 64$, istnieje n -wierzchołkowy graf*

nieskierowany G^P o maksymalnej odległości od źródła D , taki, że czas powiadomienia P na G^P wynosi $\Omega\left(n^{\frac{\log n}{\log \frac{n}{D}}}\right)$.

8.4. Grafy nieskierowane w modelu z detekcją kolizji. W tym podrozdziale prezentujemy dwa protokoły dla grafów nieskierowanych w modelu z detekcją kolizji naszego autorstwa. Pierwszy z nich to Protokół U-BOUND — dzięki niemu węzły mogą poznać D oraz ograniczenie na n . Drugi to Protokół SYNC, który przeprowadza rozgłaszanie i terminuje w czasie $\mathcal{O}(n \log n)$. Po prezentacji Protokołu SYNC krótko omówimy jego przydatność w kontekście prezentowanych wcześniej protokołów.

Protokół U-BOUND wzorowany jest na Protokole BOUND. Dodatkowo korzysta z tego, że graf jest nieskierowany (a nie tylko silnie spójny), co umożliwia poznanie ograniczenia na n szybciej, jeśli D jest małe. Przez δ_v oznaczymy $\delta(s, v)$, tj. odległość v od źródła. Choć graf jest nieskierowany, będziemy mówić o poprzednikach i następnikach, jak gdyby krawędzie były skierowane od źródła: $w \in N(v)$ jest poprzednikiem v , jeśli $\delta_w < \delta_v$, a następnikiem, jeśli $\delta_w > \delta_v$. Jeśli zaś dla sąsiadujących v i w zachodzi $\delta_w = \delta_v$, nazwiemy je *bliźniakami*.

Protokół U-BOUND działa następująco:

- (1) w czasie $\mathcal{O}(D)$ źródło wyznacza D
- (2) za pomocą Protokołu ENCODED-BROADCAST rozgłaszamy D oraz informację o odległości od źródła w czasie $\mathcal{O}(D \log D)$. Każdy węzeł poznaje D oraz swoją odległość od źródła. Dzięki temu wie, w której rundzie informacje te pozna ostatnia warstwa grafu.
- (3) wykonywany jest Protokół BOUND z dodatkowym ograniczeniem $D + 1$ na liczbę rund każdej z faz.

Poznanie D przez źródło polega na przesłaniu przez graf „fali”, która dotrze do jego końca w ciągu D rund, odbije się i wróci do źródła w czasie $\mathcal{O}(D)$. Węzeł v , który usłyszał μ po raz pierwszy w rundzie t_v , nadaje μ w rundzie $t_v + 1$. Później nadaje μ w rundzie $t_v + 1 + 3t$, jeśli usłyszał μ w rundzie $t_v + 3t - 1$ (dla $t > 0$). Jak widać, v może nadawać tylko w tych rundach j , które spełniają równość $j \equiv t_v + 1 \pmod{3}$. W ten sposób rundy przystające modulo 3 do t_v przeznaczają on na nasłuchiwanie swoich poprzedników a te przystające do $t_v + 2$ — następników, zaś bliźniaków ignoruje. Jak zwykle zakładamy, że źródło, które inicjuje protokół, odebrało μ w rundzie 0, tj. $t_s = 0$. Oczywiście zachodzi równość $t_v = \delta_v$. Poniższy lemat precyzuje zachowanie wspomnianej fali.

Lemat 3. *Węzeł v odbiera μ po raz pierwszy w rundzie δ_v i nadaje μ w rundach $t_v + 1 + 3t$ dla $t = 0, 1, 2, \dots, l(v)$, gdzie $l(v)$ to długość najdłuższej ścieżki $P = p_0, p_1, \dots, p_{l(v)}$ takiej, że $p_0 = v$ oraz p_{i+1} jest następnikiem p_i .*

Dowód. Pierwsza część lematu, czyli równość $t_v = \delta_v$ wynika wprost z tego, że każdy węzeł v nadaje μ w rundzie $t_v + 1$. Pozostałą część dowodzimy indukcyjnie względem $l(v)$:

- (1) podstawa: $l(v) = 0$ — Zgodnie z protokołem, v nada μ w rundzie $t_v + 1$. Ponieważ v nie ma następników, nie odbiera nic w rundach przystających modulo 3 do $t_v + 2$. Dlatego v nie nada μ w żadnej dalszej rundzie.
- (2) krok: $l(v) > 0$ — Przypuśćmy, że teza zachodzi dla wszystkich w takich, że $l(w) < l(v)$. Wtedy v ma następników i dla każdego w — następnika v zachodzi $l(w) < l(v)$. Stąd z założenia indukcyjnego następnicy v nie nadają μ po rundzie $t_v + 2 + 3(l(v) - 1) = t_v - 1 + 3l(v)$ a v przez to nie nadaje po rundzie $t_v + 1 + 3l(v)$. Z drugiej strony istnieje w — następnik v , który spełnia równość $l(w) = l(v) - 1$. Z założenia indukcyjnego w nadaje μ w rundach $t_v + 2, t_v + 5, \dots, t_v - 1 + 3l(v)$, więc v nadaje μ w rundach

$t_v + 4, t_v + 7, \dots, t_v + 1 + 3l(v)$. Oczywiście v nadaje μ również w rundzie $t_v + 1$.

□

```

Dane niejawne:  $G(V, E)$ , nieskierowany
Dane jawne:  $s$ 
Wynik:  $D$  oraz  $n'$ :  $\frac{n'}{2} < n \leq n'$ 

/* zakładamy, że  $s$  odebrało  $\mu$  po raz pierwszy w rundzie 0 */
for  $t = 1, 2, \dots$  do /* kolejne rundy */

    foreach  $v$  do
        if  $t = t_v + 1$  lub  $(t \equiv t_v + 1 \pmod{3}$  i  $v$  odebrał  $\mu$  w rundzie  $t - 2$ ) then
             $v$  nadaje  $\mu$ ;

/* w rundzie  $3D + 2$  źródło przestaje słyszeć  $\mu$  */
/* żaden węzeł nie nadaje a źródło dodatkowo zna  $D$  */

wartość  $D$  oraz numery warstw są rozgłaszane przy użyciu Protokołu
ENCODED-BROADCAST;

/* poniżej następuje wywołanie Protokołu BOUND z dodatkowym
ograniczeniem  $D + 1$  na długość faz */
for  $k = 1, 2, \dots$  do /* kolejne fazy */

    foreach  $v > 2^k$  do /* runda 1 fazy  $k$  */

         $v$  nadaje  $\mu$ ;

    for  $i = 2, 3, \dots, \min(2^k + 1, D + 1)$  do /* kolejne rundy fazy  $k$  */

        foreach  $v$  do
            if runda  $i - 1$  była pierwszą rundą fazy  $k$ , w której  $v$  odebrał  $\mu$  then
                 $v$  nadaje  $\mu$ ;

    foreach  $v$  do
        if  $v$  nie nadał  $\mu$  w tej fazie then
             $v$  stwierdza, że  $n' = 2^k$  i przerywa protokół

```

Protokół 16: Protokół U-BOUND

Twierdzenie 28. Protokół U-BOUND uruchomiony w dowolnym grafie nieskierowanym o n węzłach i maksymalnej odległości od źródła D , korzystając z detekcji kolizji w czasie $\mathcal{O}(D \log n)$ rozgłasza wśród węzłów wartość D oraz wyznacza n' będące potęgą liczby 2 takie, że $\frac{n'}{2} < n \leq n'$. Każdy węzeł poznaje n' oraz własną odległość od źródła a ponadto wszystkie węzły wiedzą, w której rundzie następuje koniec protokołu.

Dowód. Z Lematu 3 wynika poprawność pierwszej części protokołu, w której źródło poznaje D . Z Lematu 3 dla źródła wynika, że źródło nadaje μ po raz ostatni w rundzie $3D + 1$, ponieważ $l(s) = D$. Zatem w czasie $\mathcal{O}(D)$ źródło poznaje D . Poprawność dalszej części protokołu wynika z poprawności Protokołu BOUND, tj. Niezmiennika 1 i Twierdzenia 1 oraz możliwości skrócenia faz, kiedy wszystkie węzły znają D . Skrócenie zaczyna się dopiero w fazie $\lceil \log D \rceil$. To znaczy, że poprzednie fazy trwają w sumie $\mathcal{O}(D)$. Natomiast pozostałe $\lceil \log \frac{n}{D} \rceil + 1$ faz

ma tylko $D + 1$ rund. Sumaryczny czas dla zmienionego Protokołu BOUND wynosi $\mathcal{O}(D(\log \frac{n}{D} + 1))$. Po dodaniu czasu na poznanie D oraz rozgłoszenie D i informacji o odległości od źródła, otrzymujemy czas $\mathcal{O}(D(\log \frac{n}{D} + \log D))$, czyli $\mathcal{O}(D \log n)$. \square

Przeplatając zwykły Protokół BOUND z U-BOUND możemy zapewnić, że węzły poznają ograniczenie n w czasie $\mathcal{O}(n)$, tj. szybciej dla dużych D . Nie wiemy jak w tym samym czasie przekazać D wszystkim węzłom. Można oczywiście przestać je przekazywać — wtedy węzły będą miały świadomość, że D jest duże, tj. że $D \log D = \Omega(n)$. Jeśli celem jest poznanie ograniczenia na n , takie rozwiązanie wystarczy. Znajomość D służy tylko przyspieszeniu Protokołu BOUND.

Oczywiście można rozważać rozgłoszenie przybliżonej wartości D , kiedy okazuje się, że jest ono duże. Węzły mogą przeprowadzić kolejny protokół ENCODED-BROADCAST, z komunikatem długości $\mathcal{O}(\frac{n}{D})$. Komunikat (w tym jego długość) ustala źródło, które zna nie tylko n' , ale również D . Sama długość komunikatu może przybliżać D , jeśli wynosi dokładnie $\Theta(\frac{n}{D})$. Dodatkowo bity wiadomości mogą określać stosunek D do n dokładniej — tym dokładniej, im D jest mniejsze w stosunku do n . Jednak w skróconych komunikatach nie mieści się informacja o odległości od źródła. Dlatego równoczesne zakończenie protokołu trzeba oprzeć o wartość n' , którą wszystkie węzły poznają w tej samej rundzie.

Wspominaliśmy wcześniej, że czasem węzłom wystarczy globalny metronom zamiast globalnego zegara. Protokołowi U-BOUND, w odróżnieniu od BOUND, wystarczy metronom. Najpierw węzły budzone są podczas wyznaczania D a później poznają D oraz swoją odległość od źródła. Informacje te wystarczają do obliczenia, kiedy protokół się zakończy.

Teraz zaprezentujemy Protokół SYNC. Przez N będziemy oznaczać zbiór tych węzłów, które nie znają m , ale mają sąsiada znającego m , zaś przez A zbiór tych węzłów, które znają m i mają sąsiada w N . Pokażemy, że węzeł znający m może stwierdzić, czy należy do A . Niech wszystkie węzły znające m nadają m w rundzie r . Ich sąsiedzi odbiorą w rundzie r albo m , albo μ . Węzły, który odebrały w rundzie r sygnał μ , nadają μ w rundzie $r + 1$. Węzeł, który nadawał m w rundzie r stwierdza, że ma niepowiadomionego sąsiada wtedy i tylko wtedy, jeśli w rundzie $r + 1$ odebrał μ .

Zatem detekcja kolizji umożliwia węzłom stwierdzenie, czy należą do zbioru A , jednak pod warunkiem, że są zsynchronizowane. Podkreślamy, że korzystaliśmy z pomocy węzłów z N . Dzięki detekcji kolizji i wykorzystaniu tych węzłów potrafimy zapewnić wystarczającą synchronizację, co opiszemy później. Oczywiście zbiór N nastęrcza podobnych kłopotów co A : węzeł nieznający m musi umieć stwierdzić, czy należy do N — jak to robi, opisujemy dalej.

Węzeł, który wie, że należy do A , zaczyna nadawać by przekazać m swoim sąsiadom. Oczywiście może zagłuszać go inny węzeł z A — dlatego przez binarną selekcję wybierzemy niepusty podzbiór nadawców $S \subseteq A$, który nada m bez kolizji. Do binarnej selekcji również wykorzystamy detekcję kolizji — nie trzeba przekazywać identyfikatorów i korzystać z Protokołu ECHO. Szczegóły prezentujemy w pseudokodzie a poniżej omawiamy je bardziej przystępnie.

Protokół SYNC rozpoczyna się od wywołania Protokołu BOUND lub U-BOUND. Po zakończeniu Protokołu BOUND lub U-BOUND każdy z węzłów zna n' takie, że $\frac{n'}{2} < n \leq n'$. Dodatkowo, jeśli nawet węzły nie dysponują globalnym zegarem, od tej chwili wspólnie odliczają numery rund. Źródło nadaje m i dezaktywuje się. Dalej protokół składa się z identycznych faz. Faza składa się z $\mathcal{O}(\log n)$ rund i polega na binarnej selekcji niepustego zbioru węzłów z A , które nadadzą m bez kolizji. Przy czym przynależność węzłów do zbiorów A oraz N ustalana jest w

```

Dane niejawne:  $G(V, E)$ , nieskierowany
Dane jawne: źródło  $s$ , wiadomość  $m$ , oraz  $n'$ :  $\frac{n'}{2} < n \leq n'$ 

/*  $A$  oznacza zbiór aktywnych węzłów znajdujących  $m$  */
/*  $N$  oznacza zbiór węzłów nieznających  $m$ , które odebrały  $\mu$  na
   początku danej fazy */

wykonaj Protokół BOUND lub U-BOUND;
/* wszystkie węzły znają  $n'$  */
/* faza 1 */
 $s$  nadaje  $m$  i dezaktywuje się;
for  $k = 2, 3, \dots$  do // kolejne fazy, w każdej fazie binarna selekcja

    /* runda 1 */
    foreach  $v \in A$  do  $v$  nadaje  $m$ ;
    /* runda 2 */
    foreach  $v$ :  $v$  odebrał  $\mu$  w rundzie 1 (czyli  $v \in N$ ) do  $v$  nadaje  $\mu$ ;
    foreach  $v \in A$ :  $v$  nie odebrał  $\mu$  rundzie 2 do  $v$  dezaktywuje się;
    foreach  $v \in A$  do  $will_v := \downarrow$ ;  $x_v := 1$ ;  $y_v := n'$ ;
    for  $i := 1, 2, \dots, \lceil \log n' \rceil$  do /* kolejne tury

        /* runda 1 tury  $i$  */
        foreach  $v \in A$  do switch  $will_v$  do
            case  $\downarrow$ 
                 $(x_v, y_v) := (x_v, \lceil \frac{y_v + x_v - 1}{2} \rceil)$ ;
            case  $\uparrow$ 
                 $(x_v, y_v) := (y_v + 1, y_v + \lceil \frac{y_v - x_v + 1}{2} \rceil)$ ;
            case pass
                 $(x_v, y_v) := (1, 0)$ ,  $v$  nie robi nic więcej w tej fazie;
        if  $v \in [x_v, y_v]$  then  $v$  nadaje  $m$ ;
        /* runda 2 tury  $i$  */
        foreach  $v \in N$  do
            if  $v$  odebrał  $\mu$  w rundzie 1 tury  $i$  then  $v$  nadaje  $\mu$ ;
        foreach  $v \in A$  do
            if  $v$  odebrał  $\mu$  w tej rundzie then  $will_v := \downarrow$ ; else  $will_v := \uparrow$ ;
        // rundy 3 i 4 tury  $i$ : węzły z  $A$  deklarują swoje zamiary
        foreach  $v \in A$  do switch  $will_v$  do
            case  $\downarrow$ 
                 $v$  nadaje  $\mu$  w rundzie 3 tury  $i$ ,  $v$  nie nadaje w rundzie 4 tury  $i$ ;
            case  $\uparrow$ 
                 $v$  nie nadaje w rundzie 3 tury  $i$ ,  $v$  nadaje  $\mu$  w rundzie 4 tury  $i$ ;
        /* rundy 5 i 6 tury  $i$ : węzły z  $N$  powtarzają co odebrały w
           rundach 3 i 4 tury  $i$  */
        foreach  $v \in N$  do
            if  $v$  odebrał  $\mu$  w rundzie 3 tury  $i$  then  $v$  nadaje  $\mu$  w rundzie 5;
            if  $v$  odebrał  $\mu$  w rundzie 4 tury  $i$  then  $v$  nadaje  $\mu$  w rundzie 6;
        foreach  $v \in A$  do
            if  $will_v = \uparrow$  oraz  $v$  odebrał  $\mu$  w rundzie 5 then  $will_v := pass$ ;

```

Protokół 17: Protokół SYNC

pierwszych dwóch rundach każdej z faz i obowiązuje aż do jej końca — szczegóły dalej.

Przez binarną selekcję chcemy wybrać taki podzbiór $S \subseteq A$, by węzeł o minimalnym identyfikatorze z A należał do S . Kolejno nadają węzły ze zbioru $A \cap [x, y]$, gdzie początkowo $[x, y] = [1, n']$. Każdy węzeł z A dostaje informację zwrotną od swoich sąsiadów z N i wie, czy choć jeden z jego sąsiadów z N odebrał sygnał μ . Zerowa tura, w której nadają wszystkie węzły znajdujące m została już opisana — służy tym węzłom do stwierdzenia, czy są w zbiorze A . Te, które nie są dezaktywują się. Pozostałe zaś wiedzą, że $A \cap [x, y] \neq \emptyset$. Przedział $[x, y]$ dzielimy na dwa równe podprzedziały, które oznaczamy (lewy, prawy) jako $[x_l, y_l]$, $[x_r, y_r]$. Powtarzamy procedurę nadawania dla lewego podprzedziału. Jeśli węzeł $v \in A \cap [x_l, y_l]$ odebrał μ , wie, że w przedziale $[x_l, y_l]$ są co najmniej dwa węzły z A , więc chce przyjąć $[x, y] := [x_l, y_l]$. Jeśli zaś v nie odebrał μ , chce przyjąć $[x, y] := [x_r, y_r]$, skoro wie, że w przedziale $[x, y]$ są co najmniej dwa węzły z A a w lewym podprzedziale nie ma żadnego. Oczywiście może się zdarzyć, że v zdołał przekazać m swoim sąsiadom. Wtedy zgodnie z opisaną procedurą v chce przejść do prawego podprzedziału. Jednak to co robi v nie ma już znaczenia — nie należy już do zbioru A , o czym przekona się na początku kolejnej binarnej selekcji, a wszyscy jego sąsiedzi znają m i ignorują komunikaty nadawane przez v .

By opisana wyżej binarna selekcja działała, potrzebna jest synchronizacja — wszystkie węzły powinny decydować się na ten sam podprzedział. . . Okazuje się, że synchronizacja nie musi być pełna: jeśli węzły $v, v' \in A$ nie mają wspólnego sąsiada w N , nie zagłuszają się i nie odbierają tych samych sygnałów zwrotnych. Dlatego powiemy, że węzły $v, v' \in A$ *konkurują ze sobą*, jeśli mają wspólnego sąsiada w N . Zauważmy, że relacja konkurowania nie jest przechodnia. By binarną selekcja działała, wystarczy by zsynchronizowane były konkurujące węzły. Dokładniej, powinny one znać swoje zamiary, ponieważ v i v' mogą decydować się na różne podprzedziały. Bierze się to stąd, że oczywiście v i v' mogą mieć różne zbiory sąsiadów, przez co mogą odbierać różne sygnały. Dlatego każdy z nich deklaruje swój zamiar przy użyciu detekcji kolizji. Wtedy deklaracje te odbiera wspólny sąsiad v i v' a następnie powtarza. Dzięki temu v i v' znają swoje plany i odpowiednio je uzgadniają.

Dokładniej, każdy aktywny węzeł v dowiaduje się, czy istnieje niepusty zbiór konkurujących z nim węzłów K , który zamierza badać inny podprzedział. Jeśli $K = \emptyset$, v realizuje swój zamiar. Gdy $K \neq \emptyset$, v być może odstąpi od swego zamiaru. Ponieważ wybieramy węzeł o minimalnym identyfikatorze, v rezygnuje wtedy i tylko wtedy, gdy opowiadał się za prawym podprzedziałem. Wtedy węzły z K , które opowiadają się za mniejszymi identyfikatorami kontynuują selekcję. Tym sposobem zapewniamy, że dla każdej pary konkurujących węzłów v, v' albo oba decydują się na ten sam przedział, albo rezygnuje ten, który opowiadał się za prawym podprzedziałem. Ostatecznie węzłowi z A o najmniejszym identyfikatorze uda się nadać m bez kolizji, a być może udało się to również innym węzłom.

Do selekcji węzła z A o minimalnym identyfikatorze wystarcza $\lceil \log n' \rceil$ skróceń przedziału, na co wystarcza $6 \lceil \log n' \rceil + 2$ rund. Węzły znają n' , więc wiedzą, kiedy rozpoczynają się kolejne fazy. Węzły, które nie znają m , dzięki tej wiedzy stwierdzają, czy należą do N : należą i reagują w fazie k na sygnały μ tylko wtedy, jeśli odebrały μ w pierwszej rundzie tej fazy. W każdej fazie co najmniej jeden węzeł z A przekazuje m swoim sąsiadom z N i wiadomom, że istnieje co najmniej jeden taki sąsiad. Ponieważ faza trwa $\mathcal{O}(\log n)$ rund, czas powiadomienia wynosi $\mathcal{O}(n \log n)$ rund. Protokół SYNC można usprawnić, np. rezygnując ze ścisłego podziału na

fazy. Jednak istotna poprawa protokołu lub jego analizy wydaje się trudna. Terminacja w tym samym czasie wynika wprost z tego, że na początku każdej fazy znajdujące m węzły stwierdzają, czy należą do A i jeśli nie należą, dezaktywują się.

Twierdzenie 29. *Protokół SYNC uruchomiony w dowolnym grafie nieskierowanym o n węzłach, korzystając z detekcji kolizji dokonuje rozgłaszania oraz terminuje w czasie $\mathcal{O}(n \log n)$.*

Do przeprowadzenia Protokołu SYNC wystarcza globalny metronom (zamiast globalnego zegara), jeśli tylko korzystać w nim z U-BOUND a nie BOUND oraz wydłużyć komunikaty kontrolne (kodowane za pomocą μ i ciszy), by niepowiadomione węzły wiedziały, na które z nich reagować a które ignorować. Za to stosowanie Protokołu BOUND może skrócić przesyłane komunikaty, o czym wspominaliśmy poniżej.

Protokół SYNC działa w czasie $\mathcal{O}(n \log n)$ dla grafów nieskierowanych i to korzystając z detekcji kolizji. Ten sam czas uzyskuje Protokół SELECT-AND-SEND, który nie korzysta z detekcji kolizji. Za to SELECT-AND-SEND wysyła komunikaty, które mogą być dużo dłuższe od wiadomości m : przesyła identyfikatory węzłów, mające długość $\Theta(\log n)$. Protokół SYNC, jeśli użyć w nim Protokołu BOUND wysyła komunikaty długości $\mathcal{O}(|m|)$ (jeśli użyć Protokołu U-BOUND, długość komunikatów wzrasta do $\mathcal{O}(|m| + \log D)$). Protokół EXPLORE-AND-EXPAND, nie korzystając z detekcji kolizji, ale korzystając ze spontanicznej komunikacji gwarantuje jeszcze lepszy czas powiadomienia — $\mathcal{O}(n)$, jednak przesyłany w nim licznik również osiąga długość $\Omega(\log n)$.

Protokół SYNC działa w modelu z detekcją kolizji, dla którego nie udało nam się zdefiniować (braku) spontanicznej komunikacji. Jednak można zauważyć, że węzły, które nie poznały m nadają tylko sygnał μ i to tylko wtedy, gdy któryś z ich sąsiadów już zna m .

Jeśli $|m| = o(\log n)$, Protokół SYNC okazuje się wolniejszy od innego protokołu korzystającego z detekcji kolizji, mianowicie uważanego za niepraktyczny Protokołu ENCODED-BROADCAST, którego czas powiadomienia i terminacji wynosi $\mathcal{O}(D|m|)$.

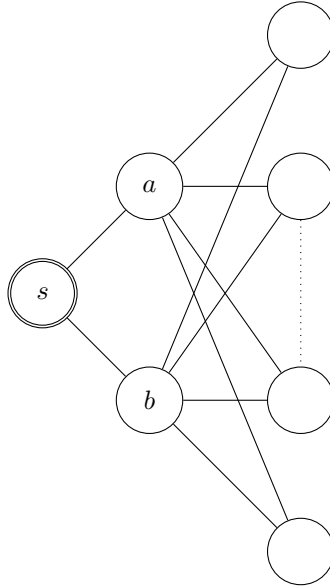
9. NIEMOŻLIWOŚĆ „ŚWIADOMEGO ROZGLĄSZANIA”

W prezentowanych protokołach węzły często dezaktywowały się, gdy wiedziały, że przekazały już wiadomość m do swoich sąsiadów. Nie były za to w stanie stwierdzić w żadnym momencie, czy rozgłaszanie już się zakończyło. W pewnych grafach niemożliwość stwierdzenia zakończenia rozgłaszania jest oczywista. Przykładem mogą być pełne grafy warstwowe: wszystkie krawędzie skierowane są od źródła, przez co żaden węzeł nie może dowiedzieć się, czy w ogóle ma następników. Pokażemy teraz, że stwierdzenie zakończenia rozgłaszania jest niemożliwe także w mniej trywialnych przypadkach. Mianowicie pokażemy, że jest to niemożliwe nawet w grafach nieskierowanych.

W prezentowanych protokołach dla grafów nieskierowanych terminacja polegała na tym, że węzły w nieskończoność czekały na sygnał nakazujący im podjęcie dalszych akcji. Jest to sytuacja odmienna od tej, w której węzeł stwierdza, że wykonał już swoje zadanie, ale problem pozostaje ten sam — nie wie, czy inne węzły zdążyły uporać się ze swymi zadaniami.

Zajmiemy się samym źródłem w grafie nieskierowanym i dla niego pokażemy, że nie może poprawnie stwierdzić, czy rozgłaszanie zostało ukończone. Wybór źródła jest zasadny: z jednej strony dowodzi, że nie jest możliwe, by każdy węzeł miał stwierdzić zakończenie rozgłaszania, z drugiej — gdyby źródło potrafiło to stwierdzić, mogłoby przez kolejne rozgłaszanie powiadomić o tym pozostałe węzły.

Protokołem świadomego rozgłaszania będziemy nazywać protokół rozgłaszania jak dotychczas, w którym dodatkowo źródło może w dowolnej rundzie stwierdzić,

RYSUNEK 3. Graf G^P .

że wszystkie węzły poznały już m . Prezentowane twierdzenie i dowód pochodzą z [CGGPR00].

Twierdzenie 30. *Dla dowolnego protokołu świadomego rozgłaszania P , mogącego korzystać ze spontanicznej komunikacji, istnieje graf nieskierowany G^P taki, że podczas wykonania P na G^P źródło s niepoprawnie stwierdza zakończenie rozgłaszania.*

Dowód. Przypuśćmy, że istnieje protokół P , który poprawnie przeprowadza świadome rozgłaszanie w dowolnym grafie nieskierowanym. Niech m' i m'' będą różnymi wiadomościami, zaś t' i t'' numerami rund, w których podczas wykonania P z wiadomością początkową odpowiednio m' i m'' w grafie składającym się wyłącznie ze źródła, s stwierdza, że rozgłaszanie jest zakończone. Niech wreszcie $t = \max(t', t'')$. Innymi słowy, t jest czasem, po którym źródło, mając do przekazania wiadomość m' lub m'' , orientuje się, że jest jedynym węzłem.

Skonstruujemy taki graf G^P mający co najmniej 2 węzły, że gdy uruchomimy na nim P z wiadomością początkową m' lub m'' , to

- (1) pewne węzły nie poznają wiadomości początkowej w ciągu t rund; oraz
- (2) w ciągu t rund źródło nie odbierze żadnego sygnału.

Warunek (1) mówi, że w ciągu t rund rozgłaszanie nie zostanie ukończone, (2) zaś, że w rundzie t źródło będzie uważało przeciwnie — na podstawie identycznego zapisu komunikacji stwierdziło ukończenie rozgłaszania w grafie, w którym było jedynym węzłem.

Graf G^P będzie miał $2^{2t} + 4t + 2$ węzłów, w tym źródło s oraz dwa szczególne węzły a i b . Dokładniej, $G^P = (X \cup \{a, b\}, \{\{a, x\}, \{b, x\} : x \in X\})$, gdzie $s \in X$. Graf G^P prezentujemy na rysunku. Chcemy, by węzły a i b przez t rund separowały s od pozostałych, tj. by przez pierwszych t rund albo jednocześnie nasłuchiwały, albo jednocześnie nadawały. Nadawać mogą różne sygnały — mają się tylko zagłuszać.

Zauważmy, że ustaliliśmy już graf, czyli topologię sieci, ale nie przydzieliliśmy jeszcze wierzchołkom identyfikatorów. Właśnie na tym polega nasze zadanie — na podstawie protokołu P tak przydzielimy identyfikatory węzłom, by warunki (1)

i (2) zachodziły. Niech źródło s ma identyfikator 1 i niech L będzie zbiorem wolnych identyfikatorów, tj. $L = \{2, 3, \dots, 2^{2t} + 4t + 2\}$. Nadamy teraz identyfikatory wybranym węzłom z X .

Niech T' będzie zbiorem tych rund z $[t]$, w których źródło nadaje sygnał, jeśli wiadomością początkową jest m' oraz źródło nie odebrało dotąd żadnego sygnału. Niech T'_1 będzie zbiorem tych rund z $[t]$, w których źródło oraz co najmniej jeden inny węzeł nadają sygnał, jeśli wiadomością początkową jest m' oraz węzły te nie odebrały dotąd żadnego sygnału. Niech A' będzie zbiorem identyfikatorów węzłów różnych od źródła, które nadają sygnały w którejkolwiek z rund z T'_1 przy powyższych warunkach, gdzie dla jednej rundy w A' jest dokładnie jeden identyfikator.

Niech T'_2 będzie zbiorem tych rund z $[t] \setminus T'$, w których dokładnie jeden węzeł (różny od źródła) nadaje sygnał, jeśli wiadomością początkową jest m' oraz nie odebrał dotąd żadnego sygnału. Niech B' będzie zbiorem identyfikatorów węzłów, które nadają sygnały w którejkolwiek z rund z T'_2 przy powyższych warunkach.

Wreszcie niech T'_3 będzie zbiorem tych rund z $[t] \setminus T'$, w których co najmniej dwa węzły (różne od źródła) nadają sygnał, jeśli wiadomością początkową jest m' oraz nie odebrały dotąd żadnego sygnału. Niech C' będzie zbiorem identyfikatorów węzłów, które nadają sygnały w którejkolwiek z rund z T'_2 przy powyższych warunkach, gdzie dla jednej rundy w C' są dokładnie dwa identyfikatory.

Zbiór $Y' = \{A' \cup B' \cup C'\}$ ma moc nie większą niż $2t$, bo zbiory A' , B' , C' są rozłączne, a na każdą rundę przypadają w nich najwyżej dwa identyfikatory. Identyfikatory z Y' przypisujemy dowolnie wybranym węzłom z $X \setminus \{s\}$. Takie przyporządkowanie oznacza, że jeśli wiadomością początkową jest m' i uda się sprawić, że węzły o identyfikatorach z Y' nie odbiorą żadnego sygnału w ciągu pierwszych t rund, niezależnie od tego jak dokończymy przypisywanie identyfikatorów, węzły a i b mogą w czasie pierwszych t rund odebrać sygnały tylko w rundach ze zbioru $(T' \setminus T'_1) \cup T'_2$ i to tylko od źródła lub węzłów z B' .

Definiujemy T'' , T''_1 , T''_2 , T''_3 , A'' , B'' , C'' , oraz Y'' analogicznie jak T' , T'_1 , T'_2 , T'_3 , A' , B' , C' , oraz Y' , tyle że dla wiadomości początkowej m'' w miejsce m' . Oczywiście $|Y''| \leq 2t$. Identyfikatory z Y'' przypisujemy dowolnie wybranym z pozostałych węzłów. Ponownie, jeśli wiadomością początkową jest m'' i uda się sprawić, że węzły o identyfikatorach z Y'' nie odbiorą żadnego sygnału w ciągu pierwszych t rund, niezależnie od tego jak dokończymy przypisywanie identyfikatorów, węzły a i b mogą w czasie pierwszych t rund odebrać jakikolwiek sygnał tylko w rundach ze zbioru $(T'' \setminus T''_1) \cup T''_2$ i to tylko od źródła lub węzłów z B'' .

Nadaliśmy identyfikatory niektórym węzłom z X (w tym s) w ten sposób, że wiemy dokładnie, które z nich, co i kiedy nadadzą w pierwszych t rundach, pod warunkiem, że węzły a i b „nie zaburzają” ich pracy, tj. w każdej rundzie będą oba nasłuchiwać albo oba nadawać. Teraz tak dobierzemy identyfikatory węzłów a i b , by faktycznie zachowywały się w opisany sposób. Wystarczy, że ich zachowanie rozpatrzemy przy założeniu, że słyszą „niezaburzoną” transmisję węzłów z $B' \cup \{s\}$ lub $B'' \cup \{s\}$, w zależności od tego, czy wiadomością początkową jest m' czy m'' . Zapis tej niezaburzonej transmisji jest z góry znany — oznaczmy go odpowiednio przez S' i S'' .

Indukcyjnie, dla wiadomości początkowej m' , definiujemy zstępujący ciąg zbiorów L'_i dla $i = 0, 1, 2, \dots, t$. Niech $L'_0 = L \setminus Y' \setminus Y''$. Wtedy $|L'_0| \geq 2^{2t} + 1$. L'_p z L'_{p-1} otrzymujemy tak: podzielmy L'_{p-1} na zbiory H_1 i H_2 , gdzie H_1 to zbiór identyfikatorów węzłów z L'_{p-1} , które w rundzie p na podstawie S' decydują się nasłuchiwać, zaś H_2 to indetyfikatory węzłów, które decydują się nadawać. Niech L'_p będzie zbiorem o większej mocy spośród H_1 i H_2 . Ostatni w ciągu zbiór L'_t ma moc nie mniejszą niż $2^t + 1$.

Analogicznie tworzymy ciąg L''_i dla $i = 0, 1, 2, \dots, t$ na podstawie S'' . Tylko $L''_0 = L'_t$. Ostatni w ciągu zbiór L''_t ma co najmniej 2 elementy. Dwie różne etykiety z L''_t przypisujemy węzłom a i b . Są one tak dobrane, że niezależnie od akcji źródła i pozostałych węzłów, w każdej z pierwszych t rund a i b albo oba nasłuchują, albo oba nadają. Stąd wszystkie pozostałe węzły w tych rundach nie odbierają żadnych sygnałów. W rundzie t , dla obu wiadomości początkowych m' i m'' , źródło stwierdza, że rozgłaszanie jest ukończone. Jednak w obu wypadkach węzły różne od a i b nie odebrały żadnych sygnałów. Więc nawet jeśli są przekonane, że znają wiadomość początkową, mylą się dla m' lub m'' . \square

Czasem, choć nie zawsze, przeprowadzenie świadomego rozgłaszania umożliwia detekcja kolizji. Wspominamy dwa przykłady z [CGGPR00], w których tak jest. Drobną modyfikacja Protokołu BOUND dla grafów silnie spójnych, połączonego z ROUND-ROBIN pozwala na świadome rozgłaszanie z niezmiennym czasem powiadomienia i terminacji $\mathcal{O}(nD)$. Również Protokół EXPLORE-AND-EXPAND dla grafów nieskierowanych, po drobnej modyfikacji przeprowadza świadome rozgłaszanie i terminuje w czasie $\mathcal{O}(n)$.

10. PROBLEMY OTWARTE

- zmniejszenie luki pomiędzy najlepszym ograniczeniem dolnym i górnym na czas rozgłaszania: $\Omega(n \log D)$ i $\mathcal{O}(n \log n \log \log n \log D)$ bądź $\mathcal{O}(n \log^2 n)$
- poprawienie rozmiarów konstrukcji selektorów lub rodzin selektywnych: ograniczenie dolne wynosi $\Omega(k \log \frac{n}{k})$
- czy dla grafów symetrycznych da się uzyskać protokoły o czasie powiadomienia $o(Dd \log \frac{n}{d} \log^3 n)$, tj. istotnie mniejszym niż w protokole BROAD-C z [CMS01] ?
- czy w protokołach BROAD-B i BROAD-C z [CMS01] lub analogicznych, istotnie zależnych od D i d da się uzyskać czas terminacji tego samego rzędu co czas powiadomienia?

LITERATURA

- [Awe85] B. Awerbuch, *A new distributed depth-first-search algorithm*; Information Processing Letters 20, pp 147–150 (1985)
- [BGV03] A. De Bonis, L. Gąsieniec, U. Vaccaro, *Generalized framework for selectors with applications in optimal group testing*; Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP), 2003, LNCS 2719, pp. 81 - 96.
- [CK05] B.S. Chlebus, D.R. Kowalski, *Almost Optimal Explicit Selectors*; 15th International Symposium on Fundamentals of Computation Theory (FCT 2005), LNCS 3623
- [CMS01] A.E.F. Clementi, A. Monti, R. Silvestri, *Selective Families, Superimposed codes, and Broadcasting in Unknown Radio Networks*; Proceedings of the 12th ACM-SIAM-SODA – 2001, pp 709-718
- [CMS03] A.E.F. Clementi, A. Monti, R. Silvestri, *Distributed broadcast in radio networks of unknown topology*; Theoretical Computer Science, 302 (2003), pp 337 - 364
- [CGGPR00] B. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, W. Rytter, *Deterministic broadcasting in unknown radio networks*; Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'00), pp 861-870
- [CGR00] M. Chrobak, L. Gąsieniec, W. Rytter, *Fast broadcasting and gossiping in radio networks*; Journal of Algorithms 43(2): pp 177-189 (2002); preliminary version in FOCS 2000
- [FF85] P. Frankl, Z. Füredi, *Forbidding just one intersection*; Journal of Combinatorial Theory Series A, 39(2), pp 172-173 (1985)
- [Ind02] P. Indyk, *Explicit constructions of selectors and related combinatorial structures, with applications*; SODA 2002: pp 697-704
- [KP02] D. Kowalski, A. Pelc, *Deterministic broadcasting time in radio networks of unknown topology*; Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS 2002)
- [KP03a] D. Kowalski, A. Pelc, *Broadcasting in undirected ad hoc radio networks*; Distributed Computing 18 (2005), pp 43-57; preliminary version in Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing (PODC'2003), pp 73-82
- [KP03b] D. Kowalski, A. Pelc, *Faster deterministic broadcasting in ad hoc radio networks*; Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'2003), LNCS 2607, pp 109-120
- [NT99] N. Nisan, A. Ta-Shma, *Extracting Randomness: A Survey and New Constructions*; Journal of Computer and System Sciences 58 (1), pp: 148 - 173 (1999)
- [Sha04] R. Shaltiel, *Recent developments in extractors*; G. Paun, G. Rozenberg, and A. Salomaa, editors, Current trends in theoretical computer science, volume 1: Algorithms and Complexity. World Scientific Publishing Co., 2004.