Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského
Bratislava

Zložitostné aspekty rádiových sietí

Diplomová práca

Diplomant: František Galčík
Vedúci diplomovej práce: RNDr. Rastislav Královič, PhD.

Bratislava, 2005

Vyhlasujem, že som diplomovú prácu vypracoval samostatne s použitím uvedenej literatúry.

Bratislava, 2005

František Galčík

Ďakujem vedúcemu diplomovej práce RNDr. Rastislavovi Královičovi, PhD. za jeho cenné rady, ochotu, pripomienky a trpezlivosť pri písaní tejto práce.

Zároveň chcem vyjadriť vďaku svojim rodičom za podporu počas písania tejto práce.

Obsah

Kapitola 1

Úvod

Komunikácia je hlavnou zložkou distribuovaných systémov. Vďaka tomu si vyslúžila pozornosť v mnohých oblastiach počítačových vied. Komunikácia je realizovaná prostredníctvom rôznych komunikačných médií. V súčasnosti má popri iných formách komunikácie rastúci význam i rádiová komunikácia. Aplikácie rádiovej komunikácie sa presunuli z oblasti tradičných vojenských aplikácií do civilnej sféry (mobilné telefóny, bezdrôtové lokálne siete – WLAN, …). Jednou z hlavných výhod rádiovej komunikácie sú relatívne nízke náklady na vytvorenie komunikačnej infraštruktúry. Okrem toho dovoľuje do istej miery aj mobilitu používateľov, čo je v niektorých aplikáciách kľúčovou požiadavkou. Požiadavka mobility používateľov so sebou prináša potrebu návrhu komunikačných protokolov tak, aby potenciálnou dynamickou zmenou štruktúry komunikačnej rádiovej siete boli minimálne (alebo vôbec) ovplyvnené. Hlavnými obmedzujúcimi faktormi rádiových sietí je limitovaný dosah použitých rádiových zariadení a vzájomná interferencia súčasného rádiového vysielania viacerých zariadení.

1.1 Základné východiská a pojmy

Rádiovou sieťou nazývame kolekciu vysielaco-prijímacích zariadení. Každé zariadenie v istom okamihu pracuje buď ako prijímač alebo ako vysielač. Zariadenia komunikujú prostredníctvom zasielania správ. Keďže dosah rádiového vysielania je obmedzený, môže zariadenie zaslať správy iba určitej podmnožine ďalších zariadení vo svojom okolí. Táto množina dosiahnuteľných zariadení závisí od vysielacej sily zariadenia a od topografickej charakteristiky okolia.

Modely rádiových sietí

Vhodným a najčastejšie používaným modelom komunikačnej rádiovej siete je graf dosiahnuteľnosti. V ňom sú jednotlivé zariadenia modelované ako vrcholy (uzly) orientovaného grafu. Tento model sa označuje aj ako grafový model. Orientovaná hrana uv v tomto grafe vyjadruje, že zariadenie v je v dosahu vysielania zariadenia u . Ak je sila vysielania u všetkých zariadení rovnaká, tak výsledný graf dosiahnuteľnosti je symetrický. To zodpovedá modelovaniu siete neorientovaným grafom dosiahnuteľnosti.

Ďalším v literatúre používaným modelom rádiovej siete je geometrický model (bližšie informácie možno nájsť v [P02] a [DP01]). Zariadenia sú v ňom reprezentované ako body v k-rozmernom euklidovskom priestore. Ku každému takémuto bodu je priradená nejaká oblasť, ktorá reprezentuje dosah rádiového vysielania daného zariadenia. Ak bod zodpovedajúci zariadeniu v leží v oblasti priradenej bodu zariadenia u, znamená to, že zariadenie v je v dosahu vysielania zariadenia u. Prirodzene najzaujímavejší je prípad 2-rozmerného euklidovského priestoru, kedy zariadenie je reprezentované bodom v rovine a oblasť priradená bodu je kruh so stredom v tomto bode. Polomer kruhu vyjadruje silu vysielania zariadenia. Geometrický model rádiovej siete je menej všeobecný a ľahko ho možno modelovať v grafovom modely. Na druhej strane však v geometrickom modely ide mnohé komunikačné problémy riešiť efektívnejšie ako vo všeobecnejšom grafovom modely.

Z pohľadu praktických aplikácií je geometrický model vhodnejší na modelovanie vtedy, keď sa zariadenia nachádzajú v približne plochých oblastiach bez veľkých prekážok a dosah vysielaní zariadení je vo všetkých smeroch rovnaký. V oblastiach s veľkými prekážkami (pohoria, stavby, ...), kde dosah rádiového vysielania závisí od smeru, je lepšie použiť modelovanie grafom dosiahnuteľnosti.

Komunikácia v rádiovej sieti

Komunikácia v rádiovej sieti prebieha v synchronizovaných kolách (časových slotoch), ktoré sú merané prostredníctvom globálnych resp. synchronizovaných lokálnych hodín. V každom z kôl sa zariadenie rozhodne, či v danom kole bude vystupovať ako vysielač alebo ako prijímač. Zariadenie vystupujúce ako vysielač vysiela správu, ktorú môžu prijať tie zariadenia, ktoré sú v dosahu vysielania a v danom kole vystupujú ako prijímače. Zariadenie vystupujúce ako prijímač prijme správu práve vtedy, ak v danom kole vysiela práve jedno zariadenie, v ktorého dosahu sa nachádza. Prijatá správa je totožná s odvysielanou správou. Ak v danom kole vysielajú prinajmenšom dve zariadenia, v ktorých dosahu sa zariadenie vystupujúce ako prijímač nachádza, hovoríme, že v prijímacom zariadení nastáva kolízia (konflikt). Z praktického pohľadu nastáva interferencia rádiových vysielaní pri prijímacom zariadení, čo má za následok znehodnotenie obsahu odvysielaných správ. V závislosti od uvažovaných schopností použitých zariadení hovoríme o sieti s detekciou kolízie, ak prijímacie zariadenie je schopné detekovať vznik kolízie. Ak sa pre prijímacie zariadenie kolízia javí rovnako, ako keď nevysiela žiadne zariadenie spomedzi tých, v ktorých dosahu sa prijímacie zariadenie nachádza, hovoríme o sieti bez detekcie kolízie.

Schopnosť rádiových zariadení odvysielať správu v rámci jedného kola všetkým zariadeniam, ktoré sú v dosahu jeho vysielania, sa javí ako vlastnosť prinášajúca významné zrýchlenie komunikácie v rádiovej sieti. Na druhej strane táto schopnosť a možný vznik kolízií počas komunikácie so sebou prináša spomalenie. Je intuitívne jasné, že možnosť vzniku kolízií robí rádiové siete ťažko koordinovateľnými a ovládateľnými. Aj vykonávanie najjednoduchších úloh tak môže byť dosť problematické. Tento efekt sa najvýraznejšie prejavuje, ak uvažujeme o distribuovanej sieti, v ktorej zariadenia dopredu nemajú žiadne informácie o topológii siete. Dôležité je preto navrhnúť také komunikačné mechanizmy, ktoré sa so vznikom kolízií dokážu vysporiadať a zároveň budú fungovať dostatočne rýchlo, aby neboli ovplyvnené potenciálne sa meniacou topológiou rádiovej siete.

Pri návrhu komunikačných mechanizmov (algoritmov, rozvrhov a protokolov) sa zvykne predpokladať, že každú správu je možné poslať a prijať iba ako celok. To znamená, že sa neuvažuje ako bude správa v konkrétnej reálnej aplikácií poslaná, či kódovaná. Súvisí to s tým, že v konkrétnych realizáciách sa správa vôbec nemusí posielať binárne kódovaná alebo k nej môže byť pridaná ďalšia informácia na zabezpečenie korektného dekódovania (napr. samoopravné kódy na potlačenie vplyvu šumu). Toto je jeden z dôvodov, prečo sa uvažuje vysielanie v synchronizovaných kolách.

Problém broadcastingu

Jednou zo základných úloh v sieťovej komunikácii, ktorou sa zaoberá aj táto práca, je broadcasting. Cieľom broadcastingu je doručiť správu (zdrojovú správu) z jedného vrcholu siete, označovaného ako zdroj, do všetkých ostatných vrcholov siete. Vzdialené vrcholy príjmu (alebo spoznajú) zdrojovú správu od iných vrcholov siete po orientovaných cestách v sieti. Predpokladá sa teda, že existuje orientovaná cesta zo zdroja do každého iného vrcholu siete. V opačnom prípade broadcasting nie je možný. Broadcasting môže byť použitý napríklad na lokalizáciu používateľa (mobilné rádiové siete), zistenie či určité zariadenie existuje (bezdrôtové lokálne siete) alebo šírenie topologickej informácie. Kľúčovou a skúmanou charakteristikou úloh vykonávaných na sieťach je celkový čas potrebný na vykonanie úlohy. Konkrétne v prípade broadcastingu na rádiových sieťach nás zaujíma počet kôl potrebných na doručenie zdrojovej správy zo zdroja do všetkých ostatných vrcholov siete.

Pri štúdiu komunikácie v rádiových sieťach sa ukázalo, že najväčší vplyv na efektívnosť distribuovaného vykonania nejakej úlohy majú schopnosti rádiového zariadenia (možnosť resp. nemožnosť detekcie kolízie) a rozsah informácie dostupnej danému zariadeniu (znalosť ohraničenia počtu zariadení v sieti, znalosť topológie časti alebo celej siete, znalosť ohraničenia počtu susedných zariadení, …). Hlavným predmetom štúdia v tejto oblasti je teda určovanie dopadu konkrétnych charakteristík zariadení na efektívnosť vykonávania úloh.

1.2 Cieľ práce

V oblasti štúdia rádiových sietí v posledných rokoch vzniklo niekoľko prác, ktoré priniesli nové, efektívnejšie algoritmy a prístupy na riešenie problému rádiového broadcastingu. Aj tento fakt ukazuje, že táto oblasť patrí k intenzívne skúmaným. Jedným z cieľov tejto práce je preto zhrnúť najaktuálnejšie výsledky, ktoré sa dosiahli pri štúdiu broadcastingu v grafovom modely rádiových sietí. Hlavným cieľom tejto práce je však navrhnúť efektívne komunikačné algoritmy na riešenie problému broadcastingu v rádiových sieťach, ktorých graf dosiahnuteľnosti má naviac nejaké špecifické topologické vlastnosti. Intuitívne sa javí, že práve použitie takýchto obmedzujúcich predpokladov na topológiu rádiovej siete, môže významným spôsobom zlepšiť dosiahnuteľnú časovú náročnosť realizácie broadcastingu.

1.3 Členenie práce

V nasledujúcej kapitole nájde čitateľ prehľad základných pojmov a princípov používaných v tejto práci spolu s formálnejšou definíciou problému broadcastingu v rádiových sieťach.

Tretia kapitola sa zaoberá rádiovým broadcastingom v sieťach, ktorých grafová topológia je všeobecná, t.j. nie sú na ňu kladené žiadne predpoklady. Uvažujeme zároveň siete s orientovaným i neorientovaným grafom dosiahnuteľnosti v modely s centralizovaným aj distribuovaným riadením. V modely s distribuovaným riadením sa osobitne venujeme sieťam s detekciou kolízie a sieťam bez detekcie kolízie. Táto kapitola je v prevažnej miere prehľadom aktuálnych výsledkov.

Štvrtá kapitola je venovaná otázke broadcastingu v sieťach, ktorých graf dosiahnuteľnosti je dvojrozmerná symetrická mriežka. Čitateľ v tejto kapitole nájde prehľad možných prístupov k riešeniu broadcastingu v takýchto grafoch dosiahnuteľnosti pri uvážení rôznych komunikačných scenárov.

Piata kapitola sa zaoberá broadcastingom v rádiových sieťach, ktorých graf dosiahnuteľnosti je planárny. Pozornosť v tejto kapitole venujeme najmä otázke centralizovaného broadcastingu.

Kapitola 2

Základné definície a pojmy

V tejto kapitole uvádzame prehľad základných definícií a pojmov, ktoré v tejto práci budeme ďalej využívať. Okrem všeobecne známych pojmov z teórie grafov bližšie zadefinujeme pojmy používané v oblasti štúdia rádiových sietí. Zároveň popíšeme základné princípy fungovania rádiovej siete s ohľadom na používaný grafový model. Prediskutujeme aj rôzne predpoklady kladené na komunikačný proces v rádiovej sieti. V záverečných dvoch podkapitolách podáme formuláciu problému broadcastingu v oboch používaných typoch riadenia: t.j. v modeloch s centralizovaným a distribuovaným riadením.

2.1 Základné pojmy z teórie grafov

Zhrnieme si základné pojmy a označenia z teórie grafov, ktoré budeme využívať v tejto práci.

Definícia 2.1.1 (Graf): *Orientovaným grafom nazývame usporiadanú dvojicu množín* $G = (V, E)$ *spĺňajúcu* $E \subseteq V \times V$. *Množinu* V *nazývame množinou vrcholov grafu a* E *množinou hrán grafu.*

Definícia 2.1.2 (Symetrický graf): *Orientovaný graf* $G = (V, E)$ *nazývame symetrickým (neorientovaným), ak platí* $(u, v) \in E \Rightarrow (v, u) \in E$.

Definícia 2.1.3 (Cesta): *Orientovaný graf* $P = (V, E)$ *nazývame cestou, ak* $V = \{x_0, \ldots, x_n\}$ *a* $E = \{(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)\}$. *Vrchol* x_0 *nazývame začiatok cesty, vrchol* x_n *nazývame koniec cesty. Počet hrán cesty je jej dĺžka.*

Definícia 2.1.4 (Vzdialenosť vrcholov): *Nech* $G = (V, E)$ *je orientovaný graf. Vzdialenosťou* $d_G(u, v)$ *dvoch vrcholov* $u, v \in V$ *grafu* G *definujeme ako dĺžku najkratšej cesty v grafe* G*, ktorej začiatok je vrchol* u *a koniec vrchol* v*. Ak žiadna taká cesta neexistuje, potom* $d_G(u, v) := \infty$.

Definícia 2.1.5 (Excentricita): *Nech* $G = (V, E)$ *je orientovaný graf. Potom excentricitu vrcholu* $v \in V$ *v grafe* $G = (V, E)$ *definujeme ako* $ex_G(v) = \max\{d_G(v, w) \mid w \in V\}$.

Definícia 2.1.6 (Vrstva grafu): *Nech $G=(V,E)$ je orientovaný graf. Potom i-tou vrstvou grafu G vzhľadom na (zdrojový) vrchol $s \in V$ nazývame množinu $L_i = \{v \in V \mid d_G(s,v)=i\}$.*

Definícia 2.1.7 (Susedia vrcholu): *Nech $G=(V,E)$ je orientovaný graf. Množinou susedov vrcholu u v grafe $G=(V,E)$ nazývame množinu $\Gamma_G(u)=\{v \in V \mid (u,v) \in E\}$*

2.2 Grafový model a komunikácia v rádiovej sieti

Ako sme už naznačili v úvodnej kapitole, v tejto práci sa budeme zaoberať výlučne grafovým modelom rádiovej siete. Opíšeme si preto základné princípy komunikácie v rádiovej sieti vzhľadom na jej graf dosiahnuteľnosti.

Definícia (Graf dosiahnuteľnosti): *Orientovaný graf $G=(V,E)$ nazývame grafom dosiahnuteľnosti rádiovej siete ak:*
1. Existuje bijekcia medzi vysielaco-prijímacími zariadeniami siete a množinou V vrcholov grafu.
2. Nech vrcholy $u,v \in V$ reprezentujú dve zariadenia v rádiovej sieti. $(u,v) \in E$ práve vtedy, ak správa vyslaná zariadením u môže dosiahnuť zariadenie v.

Dodajme, že v nasledujúcom texte tejto práce vždy, keď hovoríme o rádiovej sieti, uvažujeme tiež jej grafovú reprezentáciu, t.j. jej graf dosiahnuteľnosti.

Pri určovaní efektívnosti algoritmov sa ako charakteristiky siete berú charakteristiky grafu dosiahnuteľnosti rádiovej siete. Uvažuje sa predovšetkým počet vrcholov siete (označovaný symbolom n) a excentricita zdroja (označovaná symbolom D).

Komunikácia v rádiových sieťach prebieha prostredníctvom správ v synchronizovaných kolách (časových slotoch). Jednotlivé kolá sú určené globálnymi hodinami alebo vzájomne synchronizovanými lokálnymi hodinami. V každom kole sa zariadenie rozhodne, či v danom kole bude vystupovať ako vysielač, alebo ako prijímač. Zariadenie vystupujúce ako vysielač vysiela správu všetkým susedným zariadeniam, t.j. vrchol $u \in V$ vrcholom z množiny $\Gamma_G(u)$. Nech $S \subseteq V$ je množina vrcholov, ktoré v danom kole vystupujú ako vysielače a vysielajú nejakú správu. Nech ďalej $u \in V$ je vrchol, ktorý v grafe dosiahnuteľnosti reprezentuje zariadenie, ktoré v danom kole vystupuje ako prijímač. Definujme množinu $S_u = \{v \in S \mid u \in \Gamma_G(v)\}$. S_u je množina vrcholov (zariadení) vysielajúcich správu, ktorá dosiahne vrchol *u*. Môže nastať niekoľko prípadov:

1. Ak $|S_u|=0$, tak vrchol *u* počuje šum pozadia (background noise) a neprijíma žiadnu správu.

2. Ak $|S_u| = 1$, tak vrchol u prijíma jedinú vyslanú správu. Prijatá správa je totožná s odvysielanou správou.

3. Ak $|S_u| \geq 2$, tak hovoríme, že pri vrchole u vzniká kolízia. Vrchol u neprijíma žiadnu správu. V závislosti od použitého komunikačného scenára môže zariadenie počuť šum pozadia alebo interferenčný šum (interference noise). Ak zariadenie nemá schopnosť rozlíšiť šum pozadia a interferenčný šum, javí sa mu toto vysielanie ako vysielanie v prípade 1.

2.3 Komunikačné scenáre

Je intuitívne zrejmé, že parametre a schopnosti použitých vysielaco-prijímacích zariadení v rádiovej sieti majú najvýraznejší vplyv na návrh efektívnych komunikačných mechanizmov. V literatúre sa používa množstvo komunikačných scenárov, ktoré spoludefinujú použitý model. V prevažnej miere uvažované komunikačné scenáre vznikajú kombináciou základných predpokladov kladených na komunikačný proces v rádiových sieťach. Opíšme si najčastejšie z týchto základných predpokladov:

Randomizácia v komunikačnom procese: Použitie randomizácie v komunikačnom procese výrazne ovplyvňuje návrh algoritmov. Randomizované algoritmy realizujú úlohu broadcastingu s veľkou pravdepodobnosťou, ale nie vždy. Na druhej strane sa tieto algoritmy vyznačujú tým, že sú vo všeobecnosti rýchlejšie než deterministické algoritmy, potrebujú menej informácii o sieti, sú ľahšie implementovateľné ako distribuované a pracujú bez centrálneho monitora.

Distribuované verzus centralizované riadenie: Rozhodnutie o požiadavke centralizovaného, resp. distribuovaného riadenia má rozhodujúci význam pri každom type sieťovej komunikácie. Pri centralizovanom riadení sa predpokladá existencia monitora, ktorý má plnú znalosť siete a plánuje (riadi) vysielanie vrcholov siete. Ak vrcholy majú prístup ku globálnym hodinám je tento prístup ekvivalentný modelu s distribuovaným riadením, kde vrcholy siete plne poznajú topológiu celej siete. Vtedy sa vrcholy v každom kole riadia ma základe plánu vysielaní, ktorý si môžu vygenerovať na základe znalosti siete, zdrojového vrcholu a riadiaceho algoritmu monitora. Situácia sa výrazne komplikuje, ak vrcholy v sieti majú len limitovanú znalosť o sieti, v najextrémnejšom prípade poznajú len svoj identifikátor. Práve pre takéto siete je dôležité navrhnúť distribuované algoritmy, ktoré pracujú len na základe takejto obmedzenej informácie.

Adaptívnosť: Pri neadaptívnych algoritmoch sú všetky vysielania naplánované vzhľadom na čas už na začiatku broadcastingu. Naproti tomu adaptívne algoritmy plánujú budúce vysielanie online v závislosti na histórii prijatých správ. V prípade centralizovaného riadenia, kde sú vysielania naplánované dopredu offline, adaptívnosť nepomáha. Pomôže však napríklad v prípade distribuovaného broadcastingu, keď vrcholy poznajú topológiu celej siete, no nepoznajú zdrojový vrchol. Ak identifikátor zdrojového vrcholu je súčasťou posielanej správy, môže vrchol po prijatí prvej správy na základe znalosti zdrojového vrcholu celé ďalšie vysielanie v nasledujúcich krokoch dopredu naplánovať. Ďalším príkladom využitia adaptívnosti sú algoritmy, v ktorých vrchol môže v správe prijať informáciu o topológii nejakej vzdialenej časti siete. Túto informáciu potom môže použiť na zrýchlenie a efektívnejšie naplánovanie ďalšieho vysielania.

Možnosť / nemožnosť detekcie kolízie: Tento predpoklad umožňuje získať ďalšie informácie o sieti, ktoré je možné využiť na zefektívnenie riešenia komunikačnej úlohy. Možnosť resp. nemožnosť detekcie kolízie hovorí, čo sa stane v prípade kolízie, t.j. v prípade, keď súčasne vysiela správu viacero susedov vrcholu *v*, ktorý v danom kole vystupuje ako prijímač. Je zrejmé, že tento vrchol *v* neprijme správu od žiadneho zo susedov. Možné sú dva scenáre. Vrchol *v* nepočuje nič (okrem šumu pozadia) alebo vrchol *v* počuje interferenčný šum. Práve tieto dva scenáre sa označované ako predpoklad možnosti resp. nemožnosti detekcie kolízie. To, ktorý z nich je použitý, záleží na konkrétnych technických parametroch použitých vysielaco-prijímacích zariadení v sieti.

Odolnosť voči chybám (fault-tolerance): Väčšina algoritmov je navrhnutá tak, že predpokladá komunikačné prostredie bez výskytu chýb. To však nie je realistický predpoklad, pretože s rastúcou veľkosťou a komplexnosťou sietí rastie pravdepodobnosť zlyhania komponentov. Komunikačné algoritmy odolné voči chybám musia byť schopné splniť úlohu aj za predpokladu, že nanajvýš určitý počet komponentov siete zlyhá. Chyby, aké môžu v jednotlivých komponentoch siete vzniknúť, môžu byť rôznych druhov. Predpoklad o ich charaktere má veľký vplyv na efektívnosť konštruovaných algoritmov.

Spontánne vysielania: O spontánnych vysielaniach hovoríme v prípade, keď vrchol vysiela správu skôr než príjme prvú správu v rámci vykonávania algoritmu. Komunikácia využívajúca spontánne vysielania môže byť napríklad v prípade broadcastingu použitá na získanie informácií o okolí vrcholu ešte pred tým, než vrchol prijme zdrojovú správu. Jedným z predpokladov využitia spontánnych vysielaní je prístup ku globálnym hodinám a znalosť kola, v ktorom všetky vrcholy spoločne začnú vykonávať algoritmus. Použitie spontánnych vysielaní opäť závisí na technických parametroch použitých zariadení v rádiovej sieti.

V tejto práci sa budeme zaoberať len modelmi bez randomizácie v komunikačnom procese. Zároveň budeme predpokladať komunikačné prostredie bez chýb.

2.4 Definícia centralizovaného broadcastingu

Modely s centralizovaným riadením predpokladajú existenciu centrálneho monitora, ktorý pozná topológiu rádiovej siete a plánuje vysielania všetkých vrcholov v sieti. Vrcholy v sieti sa správajú iba na základe „povelov“ centrálneho monitora. Musia si teda pamätať len prijatú správu a byť schopné ju na príkaz monitora vyslať. Keďže topológia siete je monitoru známa, môže si celé vysielanie vypočítať a naplánovať dopredu pred začatím samotného broadcastingu. Z centralizovaného charakteru vysielania rovnako vyplýva, že jediným faktorom spomaľujúcim proces vysielania je samotný charakter rádiového vysielania – teda interferencia v dôsledku súčasného vysielania viacerých vrcholov.

Uvedený prístup s použitím centrálneho monitora je v prípade deterministických algoritmov ekvivalentný modelu s distribuovaným riadením, kde každý vrchol pozná celú topológiu siete. Ak to nie je dopredu známe, je potrebné pridať k odosielaným správam aj identifikátor zdrojového vrcholu. Po prvom prijatí správy, na základe identifikátora zdrojového vrcholu a topológie siete, vie vrchol určiť simulovaním riadiaceho algoritmu monitora aktuálne kolo prebiehajúceho algoritmu. Ďalšou simuláciou vie potom určiť kolá, v ktorých by mu monitor dal povel na vysielanie zdrojovej správy.

Definícia 2.4.1: *Nech* $G=(V,E)$ *je graf dosiahnuteľnosti rádiovej siete a* $R\subseteq V$ *je podmnožinou vrcholov grafu. Množinou vrcholov informovaných množinou R nazývame množinu* $I(R)=\{v\in V \mid \exists! x\in R, \text{ že } v\in \Gamma_G(x)\}$ *(notácia* $\exists! x$ *znamená, že existuje práve jedno x). Pre jednoprvkovú množinu* $R=\{x\}$, $I(R)=I(\{x\})=I(x)=\Gamma_G(x)$.

Definícia 2.4.2: *Nech* $G=(V,E)$ *je graf dosiahnuteľnosti rádiovej siete. Postupnosť množín vrcholov* $\Pi=(R_1,R_2,\dots,R_q)$ *, kde* $q\ge 0$ *a* $R_j\subseteq V$ *nazývame rozvrhom rádiového broadcastingu, ak* $R_{i+1}\subseteq\bigcup_{j=1}^{i} I(R_j)$ *pre každé* $i=1,2,\dots,q-1$.

Posledná definícia zahŕňa podmienku, že vrchol môže poslať správu, iba ak bol informovaný v niektorom z predchádzajúcich kôl vysielania.

Definícia 2.4.3: *Množinou vrcholov informovaných rozvrhom* $\Pi=(R_1,R_2,\dots,R_q)$ *v grafe dosiahnuteľnosti* $G=(V,E)$ *nazývame množinu* $I(\Pi)=\bigcup_{1\le i\le q} I(R_i)$ *. Pre daný graf dosiahnuteľnosti* $G=(V,E)$ *a vrchol* $s\in V$ *hovoríme, že rozvrh* Π *je prijateľný vzhľadom na* (G,s)*, ak* $R_1=\{s\}$ *a* $V\subseteq I(\Pi)$*. Dĺžku rozvrhu* $\Pi=(R_1,R_2,\dots,R_q)$ *definujeme ako* $|\Pi|=q$.

Pre daný graf dosiahnuteľnosti $G=(V,E)$ a zdrojový vrchol $s\in V$ je cieľom určiť (vygenerovať), čo najkratší prijateľný rozvrh. Na algoritmy generujúce prijateľné rozvrhy kladieme zároveň podmienku, že musia pracovať dostatočne rýchlo. Konkrétne požadujeme, aby rozvrh bol generovaný v polynomiálnom čase od počtu vrcholov siete. Poznamenajme, že čas generovania rozvrhu a dĺžka vygenerovaného rozvrhu (čas vykonávania úlohy v sieti) sú rôzne časy a navzájom priamo nesúvisia.

2.5 Definícia distribuovaného broadcastingu

Pri distribuovaných komunikačných úlohách vykonávajú všetky vrcholy v sieti rovnaký algoritmus. Ten na základe prijatých správ a informácii dostupných vrcholu riadi správanie sa vrcholu a určuje obsah ďalších vysielaných správ. Jednotlivé vrcholy siete majú len limitovanú znalosť topológie siete. V najextrémnejšom prípade je informácia v každom vrchole obmedzená na znalosť svojho identifikátora (čísla z množiny $\{1,\ldots,n\}$) , pričom vrcholy nepoznajú identifikátory svojich susedov, ani žiadne globálne parametre siete ako je veľkosť siete *n*, či excentricita zdroja *D*. Poznamenajme, že všetky uvedené výsledky ostávajú v platnosti aj za predpokladu, že identifikátor je číslo z množiny $\{1,...,r\}$, kde $r \in O(n)$. Predpoklad existencie jedinečných identifikátorov je nevyhnutný. Ak je rádiová sieť anonymná, tak deterministický distribuovaný broadcasting nemôže byť realizovaný už napríklad v 4-cykle. Návrh algoritmov, ktoré nepredpokladajú vo vrcholoch siete žiadne informácie okrem identifikátora, je dôležitý najmä pre aplikácie, kde sa topológia alebo veľkosť siete menia v priebehu času.

Na algoritmus vykonávaný vrcholmi siete sa nekladú žiadne požiadavky na časovú, či pamäťovú náročnosť. Očakáva sa však, že vrchol je schopný naplánovať si počas aktuálneho kola svoje správanie v nasledujúcom kole, či kolách.

Chýbajúce informácie o sieti vedú k problému presnej definície úlohy broadcastingu a času jeho vykonania. Pri algoritmoch centralizovaného broadcastingu je dĺžka vykonávania broadcastingu známa dopredu a teda všetky vrcholy siete vedia, kedy vykonávaný broadcasting skončí. V prípade distribuovaného broadcastingu, kde vrcholy nepoznajú globálne parametre siete, je situácia výrazne zložitejšia. Rozlišujú sa preto dve komunikačné úlohy: rádiový broadcasting (RB) a potvrdený rádiový broadcasting (ARB – acknowledged radio broadcasting). V prípade RB je cieľom doručiť zdrojovú správu do všetkých vrcholov siete. V prípade ARB je cieľom vykonať RB a naviac informovať zdroj, že RB bolo vykonané. ARB je potrebné v aplikáciách, kde zdroj má niekoľko správ na doručenie a je potrebné, aby sa vrcholy dozvedeli predchádzajúce správy skôr, než príjmu ďalšiu správu.

Predpokladá sa, že algoritmus štartuje v kole 1 a aktuálne číslo kola je určené globálnymi hodinami (tento predpoklad môžeme vynechať, ak súčasťou každej správy bude aktuálne číslo kola, vzhľadom na štartovacie kolo algoritmu a neuvažujeme spontánne vysielania).

Definícia 2.5.1: *Algoritmus vykoná rádiový broadcasting (RB) za t kôl, ak všetky vrcholy siete poznajú zdrojovú správu po kole t a žiadne správy nie sú poslané po tomto kole.*

Definícia 2.5.2: *Algoritmus vykoná potvrdený rádiový broadcasting (ARB) za t kôl, ak vykoná za t kôl RB a po kole t zdroj vie, že všetky vrcholy poznajú zdrojovú správu.*

Kapitola 3

Broadcasting vo všeobecných grafoch

Otázka konštrukcie algoritmov realizujúcich efektívny rádiový broadcasting v sieťach, na ktorých graf dosiahnuteľnosti nie sú kladené žiadne topologické predpoklady – t.j. v sieťach, ktorých grafy dosiahnuteľnosti sú všeobecné grafy, patrí k najintenzívnejšie skúmaným otázkam v tejto oblasti. V posledných desiatich rokoch sa dosiahlo mnoho výsledkov, ktoré postupne znižujú horný a zvyšujú dolný odhad počtu kôl potrebných na realizovanie broadcastingu. Táto kapitola podáva predovšetkým prehľad súčasného stavu problematiky prostredníctvom prehľadu známych výsledkov. Kapitola je rozdelená do 3 podkapitol. Prvá sa venuje broadcastingu v sieťach s centralizovaným riadením. Druhá podkapitola sa zaoberá otázkou broadcastingu v sieťach s distribuovaným riadením bez možnosti detekcie kolízie. Tretia podkapitola sa naopak zaoberá otázkou broadcastingu v sieťach s distribuovaným riadením s možnosťou detekcie kolízie. V tretej podkapitole zároveň prezentujeme náš algoritmus, realizujúci rádiový broadcasting v distribuovaných sieťach s detekciou kolízie bez využitia spontánnych vysielaní v čase $O(n + D \log D)$ kôl.

3.1 Centralizovaný broadcasting

Ako je ukázané v práci [CK85], nájdenie najkratšieho prijateľného rozvrhu pre daný graf a zdrojový vrchol je NP-úplný problém. Preto je potrebné nájsť algoritmy bežiace v polynomiálnom čase na konštrukciu prijateľných rozvrhov, ktoré budú čo najlepšie aproximovať optimálny čas broadcastingu.

Tvrdenie 3.1.1 [CW87]: *Nech $G = (V, E)$ je orientovaný graf dosiahnuteľnosti rádiovej siete. Potom existuje deterministický algoritmus, ktorý v polynomiálnom čase pre daný zdrojový vrchol s vygeneruje prijateľný rozvrh Π taký, že $|\Pi| \in O(D \log^2 n)$.*

Idea dôkazu: Základom algoritmu je procedúra *SEA* (*Spokesman election algorithm*). Nech *X* je množina vrcholov, ktoré prijali zdrojovú správu a nech $Y \subseteq \Gamma_G(X)$ je množina tých susedných vrcholov množiny *X*, ktoré ešte neprijali zdrojovú správu. Algoritmus *SEA* skonštruuje v polynomiálnom čase takú množinu $S \subseteq X$ (*S* je nazývané množinou hovorcov), že $|I(S)| > \frac{Y}{\ln X}$. Voľba množiny S teda garantuje, že pri vysielaní vrcholov množiny *S*, prinajmenšom $\frac{1}{\ln X}$-tina neinformovaných susedov množiny *X* prijme zdrojovú správu. Samotný algoritmus generovania rozvrhu pracuje po vrstvách. Nech všetky vrcholy vrstvy L_{i-1} sú informované. Potom ak algoritmus *SEA* aplikujeme nanajvýš $\ln|L_{i-1}|\ln|L_i|$ krát na množinu vrcholov L_{i-1}, máme z vlastností algoritmu *SEA* zaručené, že všetky vrcholy vrstvy L_i sa stanu informovanými. Celková dĺžka generovaného rozvrhu je tak $O(D\log^2 n)$, pričom na prechod do nasledujúcej vrstvy je potrebných $O(\log^2 n)$ kôl. ♣

V práci [KPb04] Kowalski a Pelc prezentovali algoritmus generujúci rozvrhy dĺžky $O(D\log n + \log^2 n)$. Tento algoritmus nahradil dovtedy v mnohých iných broadcastovacích algoritmoch používaný randomizovaný algoritmus, ktorý generoval rozvrhy asymptoticky rovnakej dĺžky. Tým sa tieto algoritmy, ktorých randomizovanými robilo iba použitie spomínanej randomizovanej procedúry, stali nerandomizovanými.

Tvrdenie 3.1.2 [KPb04]: *Nech $G=(V,E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete. Potom existuje deterministický algoritmus, ktorý v polynomiálnom čase pre daný zdrojový vrchol $s \in V$ vygeneruje prijateľný rozvrh Π taký, že $|\Pi| \in O(D\log n + \log^2 n)$.*

Idea dôkazu: Algoritmus generuje rozvrh vo fázach. V rámci každej fázy generuje množiny vysielajúcich vrcholov vo všetkých vrstvách. Aby sa vyhlo interferencii pri súčasnom vysielaní v susedných vrstvách, je každá fáza rozdelená na 3 podfázy. V každej z nich sa pracuje súčasne s vrstvami, ktoré sú od seba oddelené 2 ďalšími vrstvami. Ak aj teda nastane súčasné vysielanie v rôznych vrstvách, nikdy nedôjde k vzájomnému ovplyvňovaniu sa. S vrstvou L_j vo fáze *k* sú asociované množiny $S_{k,j} \subseteq \bigcup_{i>j} L_i$ a $R_{k,j} \subseteq L_j$. $S_{k,j}$ je množina tých vrcholov vo vrstvách nasledujúcich vrstvu L_j, ktorým majú vrcholy vrstvy L_j zabezpečiť doručenie zdrojovej správy. Hovoríme, že vrchol *w* je predchodcom vrcholu *v*, ak vrchol *w* leží na nejakej najkratšej ceste zo zdroja do vrcholu *v*. $R_{k,j}$ je taká množina už informovaných vrcholov vrstvy L_j vo fáze *k*, že pre každý vrchol z $v \in S_{k,j}$ je v množine $R_{k,j}$ aspoň jeden jeho predchodca $w \in L_i$. Na začiatku algoritmu $S_{1,1} = \bigcup_{i>1} L_i$ a $R_{1,1} = \emptyset$. Všetky ďalšie množiny $S_{k,j}$ a $R_{k,j}$ sú prázdne. Vrcholy množiny $S_{k,j}$ a $R_{k,j}$ vytvárajú bipartitný graf $G_{k,j} = (V_{k,j}, E_{k,j})$, kde $V_{k,j} = S_{k,j} \cup R_{k,j}$ a $(w,v) \in E_{k,j}$ práve vtedy, ak vrchol $w \in R_{k,j} \subseteq L_j$ je predchodcom vrcholu $v \in S_{k,j}$.

Aplikujme v grafe $G_{k,j} = (V_{k,j}, E_{k,j})$ algoritmus *SEA* z predchádzajúcej vety $2\log n$ krát, pričom množina informovaných vrcholov pre algoritmus *SEA* je $R_{k,j}$. Získame tak rozvrh, ktorý zabezpečí, že prinajmenšom polovica vrcholov v $S_{k,j}$ virtuálne prijme správu. Nech $S'_{k,j} \subseteq S_{k,j}$ je množine tých vrcholov, ktoré virtuálne v zmysle algoritmu *SEA* príjmu správu. Graf $G_{k,j} = (V_{k,j}, E_{k,j})$ nie je podgrafom grafu $G=(V,E)$ a tak virtuálne prijatie neznamená aj reálne prijatie správy. Virtuálne prijatie správy vrcholom $v \in S_{k,j}$ však pri reálnom vysielaní podľa vygenerovaného rozvrhu znamená, že aspoň jeden jeho predchodca vo vrstve L_{j+1} správu skutočne prijme. Rovnako aj v prípade, že $v \in L_{j+1}$ dôjde k reálnemu prijatiu správy. Výsledkom tejto operácie je to, že požiadavku (kladenú na vrstvu L_j) na zabezpečenie doručenia zdrojovej správy vrcholom množiny $S'_{k,j}$ môžeme preniesť na vrcholy vrstvy L_{j+1}. Teda vrcholy množiny $S'_{k,j}$ presunieme z množiny $S_{k,j}$ do množiny $S_{k+1,j+1}$. Ak po tejto operácie je množina $S_{k,j}$ dostatočne malá ($|S_{k,j}| \leq 4\log n$) a máme zaručené, že nové prvky do nej už z predchádzajúcich vrstiev nebudú pribúdať (ako je ukázané, toto nastane keď $j = (k+1) - \log n$), tak aplikovaním algoritmu *RoundRobin* (pre každý vrchol, ktorý ešte neprijal správu, určíme jeden jeho virtuálny susedný vrchol, ktorý bude správu sám v určitom kole vysielať) získame, že všetky vrcholy $S_{k,j}$ príjmu virtuálne zdrojovú správu. Množina $\bigcup_{i=1}^{j} S_{k+1,i}$ tak bude prázdna, čo znamená, že všetky vrcholy vo vrstvách $\bigcup_{i=1}^{j} L_i$ prijali zdrojovú správu. Po $D + \log n$ fázach algoritmu tak zdrojovú správu príjmu všetky vrcholy siete. Celková dĺžka vygenerovaného rozvrhu je $O((D + \log n).\log n) = O(D \log n + \log^2 n)$. ♣

Ako vyplýva z výsledkov v prácach [EKa04] a [EKb04] rozvrh produkovaný uvedeným algoritmom má optimálny rád magnitúdy za predpokladu, že $NP \subset BPTTIME(n^{O(\log\log n)})$.

Poznamenajme, že randomizovaný algoritmus na generovanie rozvrhu dĺžky $O(D \log n + \log^2 n)$ bol autormi Bar-Yehudom, Goldreichom a Itaiom prezentovaný v práci [BGI92]. Tento randomizovaný algoritmus bol pôvodne použitý ako podprocedúra pri konštrukcii algoritmov (z tvrdení 3.1.3 a 3.1.4) na generovanie rozvrhov dĺžky $O(D + \log^4 n)$ a $O(D + \log^5 n)$. Tieto samotné algoritmy, ako uvidíme neskôr, boli deterministické, no použitie randomizovanej podprocedúry ich robilo tiež randomizovanými. Ich nerandomizované verzie s použitím vtedy najrýchlejšieho deterministického algoritmu generovali rozvrhy dĺžky $O(D + \log^5 n)$ a $O(D + \log^6 n)$. Algoritmus má aj niekoľko zaujímavých vlastnosti, ktoré vyplývajú z využitia randomizácie: nepredpokladá jedinečné priradenie identifikátorov, funguje aj pre orientované grafy dosiahnuteľnosti, je jednoduchý a ľahko prispôsobivý zmene topológie siete. Jedinými vstupnými parametrami algoritmu sú počet vrcholov v sieti a pravdepodobnosť ε neúspešného vykonania broadcastingu. Z charakteru algoritmu tiež vyplýva, že môže byť za splnenia požadovaných vstupných predpokladov prevedený aj do distribuovanej formy. Algoritmus generuje rozvrh pracujúci vo fázach. Základnou ideou algoritmu je náhodným rozhodovaním („hádzanie mincou“ o tom, či pokračovať alebo nepokračovať vo vysielaní prijatej správy v rámci danej fázy) zabezpečiť, že každým kolom fázy sa množina susedov, ktoré chcú vysielať zdrojovú správu vzhľadom na nejaký konkrétny vrchol zníži na polovicu. Po $\log(n/\varepsilon)$ fázach je s pravdepodobnosťou presahujúcou $1-\varepsilon$ vykonaný úspešný broadcasting. Dĺžka generovaného rozvrhu je $O((\log \frac{n}{\varepsilon} + D)\log n)$.

V práci [GM95] Gaber a Mansour prezentovali algoritmus generujúci rozvrhy dĺžky $O(D+\log^5 n)$. I keď v súčasnosti existuje algoritmus generujúci kratšie rozvrhy, pozrieme sa na tento algoritmus dôkladnejšie, pretože algoritmus (z tvrdenia 3.1.4) generujúci rozvrh dĺžky $O(D+\log^4 n)$ vychádza a je modifikáciou myšlienky práve tohto algoritmu.

Tvrdenie 3.1.3 [GM95]**:** *Nech $G=(V,E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete. Potom existuje deterministický algoritmus, ktorý v polynomiálnom čase pre daný zdrojový vrchol s vygeneruje prijateľný rozvrh Π taký, že $|\Pi| \in O(D+\log^5 n)$.*

Idea dôkazu: Základnou myšlienkou tohto algoritmu je rozdeliť graf vhodným spôsobom na clustre. Tieto clustre nemusia byť disjunktné, no musia spĺňať isté podmienky týkajúce sa ich priemeru a stupňa disjunktnosti. Clustre sú konštruované tak, že umožňujú vytvorenie stromu – t.j. pre každý cluster (s výnimkou koreňového) je určený práve jeden iný cluster, ktorý je v zmysle stromu jeho rodičom. Cluster obsahuje jeden významný vrchol - **messanger**, cez ktorý do clustra prichádza zdrojová správa z rodičovského clustra. Okrem toho je v clustry skupina vrcholov - reprezentantov, z ktorých sa správa šíri do synovských clustrov. Reprezentanti reprezentujú synovské clustre v ich rodičovskom clustry. Reprezentant synovského clustra je hranou spojený s messengerom tohto synovského clustra. Z týchto reprezentantov je jeden význačný – zvolený reprezentant**.** Broadcastovací rozvrh je postavený tak, aby sa zdrojová správa dostala z messengera do zvoleného reprezentanta čo najrýchlejšie po najkratšej ceste. Pri broadcastovaní paralelne (v rôznych časových slotoch) bežia 3 úlohy:

- *Broadcast-All*: doručuje správu z messangera do všetkých ostatných vrcholov clustra
- *Broadcast-Through*: doručuje správu z messangera do zvoleného reprezentanta po najkratšej ceste
- *Group-to-Group*: doručuje správu z reprezentanta do messengera synovského uzla

Konštrukcia clustrov:

Rozdeľme vrcholy podľa vzdialenosti od zdrojového vrcholu (príslušnosti k vrstve) do *x* skupín: $G_i = \{v \in V(G) \mid v \in L_j, (i-1).D/x+1 \le j \le i.D/x\}$. Teda skupina G_i pozostáva z vrcholov vo vzdialenosti $(i-1).D/x+1$ až $i.D/x$ od zdroja. Pre každý vrchol $u \in L_{(i-1).D/x+1}$ z najnižšej vrstvy skupiny G_i definujeme množinu $S_u^{(i)} = \{v \in G_i \mid u \textit{ leží na nejakej najkratšej ceste zo zdroja do vrcholu } v\}$. Inými slovami je $S_u^{(i)}$ množina tých vrcholov skupiny G_i, do ktorých sa dá dostať takými cestami, na ktorých je každý vrchol v inej vrstve a obsahujú vrchol *u*. Clustre budeme vytvárať ako zjednotenia niekoľkých množín $S_u^{(i)}$.

Definícia: *Zafarbením clustrov* $C_1,\dots,C_m$ *použitím* α *farieb nazývame také priradenie farby (z množiny* $\{1,\dots,\alpha\}$*) každému clustru, že neexistuje hrana spájajúca dve clustre rovnakej farby, t.j.* $\forall v_i \in C_i, v_j \in C_j, i \neq j$ *platí, že ak* $(v_i, v_j) \in E$*, potom* $color(C_i) \neq color(C_j)$.

Lema: *Pre dané množiny* $S_1^{(i)},\dots,S_m^{(i)}$ *existujú clustre* $C_1^{(i)},\dots,C_k^{(i)}$ *s nasledovnými vlastnosťami:*

1. *Každé* C_j *je zjednotením niekoľkých množín* S_r
2. *Každé množina* S_r *sa nachádza v aspoň jednom clustry* C_j
3. *Priemer každého clustra* C_j *je nanajvýš* $O(D/x.\log n)$
4. *Existuje* α *zafarbenie clustrov* C_j *také, že* $\alpha \in O(\log n)$
5. *Clustre sa dajú skonštruovať v čase* $O(m \log n)$

Vďaka tomu, že clustre boli konštruované ako zjednotenia množín S_r, máme zaručené, že pre každý vrchol $u \in C_j^{(i)}$ z najvyššej vrstvy skupiny G_i existuje vrchol $v \in C_j^{(i)}$ v najnižšej vrstve taký, že celá najkratšia cesta z *u* do *v* leží v $C_j^{(i)}$.

Nech $C_i = \bigcup_j C_j^{(i)}$ je množina clustrov skupiny G_i pre $i = \{1,\dots,y\}$, kde y je počet skupín. Nasledujúci algoritmus priradí každému clustru C jeho ohodnotenie $rank(C)$, skonštruuje BFS kostru τ z clustrov a v každom clustry určí messengera a reprezentantov v clustroch predchádzajúcej skupiny.

Algoritmus inicializuje strom τ na prázdny strom. Priradí ohodnotenie „rank“ 0 každému clustru $C \in C_y$. V každom clustry $C \in C_y$ ďalej vyberie ľubovoľný vrchol $w \in C$ z najvyššej vrstvy skupiny G_y ako messangera clustra *C*. Zároveň zvolí ľubovoľný susedný vrchol messengera z predchádzajúcej vrstvy ako reprezentanta clustra *C*. Teda, ako ukážeme aj neskôr, reprezentant clustra $C' \in C_j$ je vždy nejaký vrchol z najvyššej vrstvy nejakého clustra $C'' \in C_{j-1}$. Pritom platí, že messanger clustra a jeho reprezentant sú spojený hranou v grafe dosiahnuteľnosti. Algoritmus inicializuje množinu U nepokrytých clustrov rovnú množine clustrov C_y ($U := C_y$).

Algoritmus po tom, čo spracuje clustre skupiny G_j pokračuje spracúvaním clustrov v skupine G_{j-1} pre $j = y,\dots,2$, t.j. pokračuje spracúvaním clustrov množiny C_{j-1} v ľubovoľnom poradí. Pre každý cluster $C \in C_{j-1}$ robí nasledujúce: Nech $N \subseteq U \subseteq C_j$ je množina tých clustrov, ktorých reprezentant patrí do clustra *C*. Pre každý cluster $C' \in N$ algoritmus pridá do stromu τ hranu (C, C'). Cluster *C* sa stane rodičom clustrov z množiny *N*. Nech $r = \max\{rank(C') \mid C' \in N\}$ je najvyššie ohodnotenie niektorého z clustrov množiny *N*. Ak existuje jediný cluster $C' \in N$ s ohodnotením *r*, tak reprezentant clustra C' sa stane zvoleným reprezentantom v clustry C a cluster C sa ohodnotí hodnotou r ($rank(C) := r$). V opačnom prípade sa cluster C ohodnotí hodnotou $r+1$ ($rank(C) := r+1$) a ako zvolený reprezentant v clustry C sa vyberie ľubovoľný reprezentant clustra s ohodnotením *r*. Ako messanger clustra C sa vyberie ľubovoľný vrchol z najnižšej vrstvy clustra, ktorý leží na najkratšej ceste zo zdroja k zvolenému reprezentantovi clustra C – t.j. ako messanger sa vyberie ľubovoľný predchodca zvoleného reprezentanta. Vzhľadom na to, že cluster C vznikol zjednotením množín $S_i^{(j-1)}$, taký vrchol vždy existuje. V prípade, že množina N je prázdna, nastaví sa ohodnotenie clustra C na 0 a ako messanger sa vyberie ľubovoľný vrchol z najnižšej vrstvy clustra C. Za reprezentanta clustra C sa vyberie ľubovoľný susedný vrchol messengera z predchádzajúcej vrstvy (teda vrchol zo skupiny G_{j-2}). Množina nepokrytých clustrov sa nastaví na $U := U - N$. Toto sa opakuje, až kým sa nespracujú všetky clustre množiny C_{j-1}. Po skončení sa nastaví $U := C_{j-1}$ a pokračuje sa spracovávaním clustrov množiny C_{j-2}. Na konci algoritmu je skonštruovaný strom clustrov a pre každý cluster je definovaný jeho reprezentant a messanger.

Broadcastovací rozvrh je výsledkom už spomínaných 3 úloh (procedúr), ktoré bežia v rámci každého clustra: *Broadcast.-Though*, *Broadcast-All* a *Group-to-Group*. Jednotlivé úlohy bežia v rôznych časových kolách (modulo 3), čo garantuje, že sa navzájom neovplyvňujú.

<u>*Broadcast-Through*</u> – cieľom tejto úlohy je doručiť správu z messangera do zvoleného reprezentanta. Po tom, čo messanger prijme správu, čaká maximálne $O(\log n)$ kôl na kolo, ktoré je štartovacie pre danú farbu clustra. V tomto kole sa začína vysielať správa po najkratšej ceste k zvolenému reprezentantovi. Vďaka tomu, že sa vysielanie začína až v kole určenom pre danú farbu clustra a vysielanie je realizované po najkratšej ceste (čiže po vrstvách), je zabezpečené, že jednotlivé *Broadcast-Through* úlohy susedných clustrov sa navzájom nebudú ovplyvňovať. Poznamenajme tiež, že jeden vrchol môže byť súčasne vo viacerých clustroch. Každý z nich má však priradenú inú farbu.

<u>*Broadcast-All*</u> – cieľom tejto úlohy je doručiť správu z messangera do všetkých ostatných vrcholov v clustry, teda aj k reprezentantom, ktorí neboli zvolení. Ako broadcastovací algoritmus môžeme použiť deterministický algoritmus generujúci rozvrh dĺžky $O(D\log n + \log^2 n)$. Poznamenajme, že v pôvodnej verzii algoritmu sa na účel broadcastingu v rámci clustra využíval randomizovaný algoritmus z [BGI92]. Vzhľadom na to, že priemer každého clustra je $O(D/x.\log n)$, bude broadcasting v clustry vyžadovať najviac $O(D/x.\log^2 n)$ kôl. Messanger po prijatí správy čaká na kolo, ktoré je začiatkom vykonávania broadcastovacieho algoritmu. Toto kolo je každé $O(D/x.\log^2 n)$ - té kolo. Messangery rôznych clustrov, ktoré prijali zdrojovú správu v rovnakej časovej perióde *Broadcast-All*, tak v rámci skupiny začínajú vykonávanie broadcastingu v rovnakom kole. Broadcasting je realizovaný bez ohľadu na farbu clustra. Pri broadcastovacích algoritmoch sa predpokladá jeden zdrojový vrchol. V tomto prípade, ich však môže byť niekoľko, čo vyžaduje čiastočnú modifikáciu algoritmu. Presnejšie na broadcasting v clustry je treba pozerať sa ako na broadcasting v grafe dosiahnuteľnosti, kde existuje „virtuálny“ zdrojový vrchol a messengery clustrov, ktoré v danej perióde prijímajú správu, sú jeho susednými vrcholmi. Tento graf skupiny v danej perióde doplnený o virtuálny vrchol má priemer rádovo

rovnaký ako je priemer každého z clustrov. Aby sa vyhlo interferencii *Broadcast-All* algoritmov bežiacich v susedných skupinách, rozdelíme vykonávanie tejto úlohy do 2 striedajúcich sa typov kôl: kolo pre párne skupiny a kolo pre nepárne skupiny.

<u>*Group-to-Group*</u> – cieľom tejto úlohy je doručiť zdrojovú správu z reprezentanta clustra do messangera clustra. Ide o doručenie správy medzi dvomi clustrami, ktoré sú v rôznych skupinách. Reprezentant clustra, rovnako ako v predchádzajúcich úlohách, čaká na kolo, ktoré je začiatkom danej periódy vykonávania úlohy. V tomto prípade čaká nanajvýš $O(\log^2 n)$ kôl. Na doručenie správy sa využíva algoritmus *SEA* (z tvrdenia 3.1.1), ktorý doručí zdrojovú správu počas nanajvýš $O(\log^2 n)$ kôl. Opäť všetci reprezentanti začínajú vykonávať algoritmus v rovnakom kole v rámci danej periódy.

Lema: *Pre ľubovoľný strom T s n vrcholmi platí, že* $0 \le rank(T) \le \log n$.

Broadcastovací rozvrh je postavený tak, že zdrojová správa sa k vrcholu cez clustre dostáva tým rýchlejšie, čím je na ceste k nemu viac zvolených reprezentantom. Zvolený reprezentanti sú však konštruovaný na základe ohodnotenia „rank“ clustrov. Z predchádzajúcej lemy však vyplýva, že na ľubovoľnej ceste v clustrovom strome sa rank mení nanajvýš $\log n$ krát. Teda na ceste k ľubovoľnému vrcholu sa zdrojová správa dostáva cez nanajvýš $\log n$ procedúr *Broadcast-All*. Celkový dĺžka rozvrhu vygenerovaného algoritmom je $O(D + x.\log^2 n + (D / x.\log^2 n)\log n)$. Pre $x = \log^3 n$ dostávame dĺžku vygenerovaného rozvrhu $O(D + \log^5 n)$. ♣

Uvedený algoritmus bol Elkinom a Kortsarzom v práci [EKc04] vylepšený na generovanie rozvrhov dĺžky $O(D + \log^4 n)$.

Tvrdenie 3.1.4 [EKc04]**:** *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete. Potom existuje deterministický algoritmus, ktorý v polynomiálnom čase pre daný zdrojový vrchol s vygeneruje prijateľný rozvrh* Π *taký, že* $|\Pi| \in O(D + \log^4 n)$.

Idea dôkazu: Uvedený algoritmus do značnej miery vychádza z algoritmu z predchádzajúceho tvrdenia 3.1.3. Na rozdiel od predchádzajúceho algoritmu nie sú skupiny G_i disjunktné. Konkrétne platí, že $G_i \cap G_{i+1} = L_j$. Teda pre dve za sebou idúce skupiny G_i a G_{i+1} platí, že najnižšia vrstva G_{i+1} je najvyššou vrstvou G_i. Zároveň aj clustre v nasledujúcich skupinách nie sú disjunktné a majú spoločné vrcholy v spomínanej spoločnej vrstve dvoch nasledujúcich skupín. Cieľom tejto úpravy je odstrániť potrebu vykonávania procedúry *Group-to-Group*. Ďalším dôsledkom tejto úpravy je to, že reprezentant clustra už nie je v inom clustry, ale messanger je zároveň reprezentantom clustra. Algoritmus nevyužíva *RoundRobin* prístup rozdelenia kôl pre jednotlivé úlohy pri konštrukcii broadcastovacieho plánu. Plán je konštruovaný s využitím rekurzívnej procedúry *Period*, ktorá pre dané kolo t a danú množinu clustrov S, ktorých reprezentanti sú v danom kole informovaný, konštruuje broadcastovací plán. Tento broadcastovací plán zabezpečí doručenie zdrojovej správy do všetkých clustrov, ktoré majú byť reprezentantmi (messangermi) clustrov v množine S informované.

Pre daný graf dosiahnuteľnosti $G=(V,E)$ a zdrojový vrchol s je celková dĺžka generovaného rozvrhu $D+O(\sqrt{D}.\log^2 n)=O(D+\log^4 n)$. ♣

Všimnime si, že väčšina uvedených algoritmov pracuje s neorientovanými grafmi dosiahnuteľnosti. Dôvod, prečo tie algoritmy nie je možné preniesť na orientované grafy, je ten, že algoritmy pracujú súčasne s vrcholmi vo viacerých vrstvách a predpokladajú, že ak 2 vrcholy ležia v dostatočne vzdialených vrstvách, tak nehrozí, že by ich spoločné vysielanie v rovnakom kole viedlo k interferencii. Tento predpoklad však pre orientované grafy neplatí.

V ďalšej časti sa pozrieme na dolné odhady centralizovaného broadcastingu. Dodajme, že triviálnym dolným odhadom dĺžky rozvrhu rádiového broadcastingu je excentricita zdroja $\Omega(D)$.

Tvrdenie 3.1.5 [ABLP91]: *Existuje množina neorientovaných grafov dosiahnuteľnosti o n vrcholoch s polomerom 2 takých, že každý prijateľný rozvrh rádiového broadcastingu je dĺžky* $\Omega(\log^2 n)$.

Poznamenajme len, že dôkaz predchádzajúceho tvrdenia 3.1.5 je založený na pravdepodobnostnom dôkaze existencie množín s istými vlastnosťami.

Z predchádzajúcich tvrdení vyplýva, že pre neorientované grafy dosiahnuteľnosti $G=(V,E)$ s priemerom $D\in O(\log^2 n)$ a $D\geq k\log^4 n$ existujú algoritmy, ktoré generujú prijateľné rozvrhy rádiového broadcastingu asymptoticky optimálnej dĺžky $O(D)$. Existencia optimálnych prijateľných rozvrhov pre zvyšnú triedu grafov ostáva otvorený problém.

3.2 Distribuovaný broadcasting bez detekcie kolízie

V tejto podkapitole popíšeme sériu algoritmov, ktoré realizujú rádiový broadcasting v sieťach, ktorých graf dosiahnuteľnosti je orientovaný. Uvedené algoritmy nevyužívajú spontánne vysielania (pokiaľ nie je uvedené inak), t.j. vrchol v sieti vysiela až po tom, čo prijme nejakú správu ako dôsledok vykonávania algoritmu.

Tvrdenie 3.2.1 [CGGPR00]: *Nech* $G = (V, E)$ *je orientovaný graf dosiahnuteľnosti rádiovej siete a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny algoritmus, ktorý realizuje rádiový broadcasting (RB) za* $O(n^2)$ *kôl.*

Idea dôkazu (Algoritmy Round-Robin a Simple-Sequencing): Algoritmus *Round-Robin* je základným algoritmom pre distribuovaný rádiový broadcasting. Algoritmus predpokladá, že každý vrchol pozná veľkosť siete *n*, resp. *r* horné ohraničenie identifikátorov v sieti.

Algoritmus *Round-Robin* pracuje v *n* identických fázach, z ktorých každá pozostáva z *n* kôl. V každej fáze všetky vrcholy, ktoré prijali zdrojovú správu vysielajú v niektorom z kôl fázy. Konkrétne, vrchol s identifikátorom *i* vysiela zdrojovú správu v *i*-tom kole fázy.

V prípade, že parameter *n* je neznámy, môžeme použiť následovnú techniku, ktorá sa v obmenách využíva v prípadoch, kde parameter *n* je neznámy, no základné verzie broadcastovacích algoritmov takúto znalosť vyžadujú. Túto techniku však nie je možné použiť vždy.

Algoritmus *Simple-Sequencing* pracuje vo fázach. Vo fáze *k* vykonávajú algoritmus *Round-Robin(* 2^k *)* všetky vrcholy s identifikátormi $1, \ldots, 2^k$ s nasledovnou modifikáciou: vrchol, ktorý prijal zdrojovú správu a vysielal ju už v rovnakom kole fázy, nevysiela v žiadnom z nasledujúcich kôl fázy.

Po $\lceil \log n \rceil$ fázach príjmu všetky vrcholy zdrojovú správu. Fáza *k* pozostáva z 4^k kôl. Celkovo je teda na vykonanie broadcastingu i pri neznalosti parametra *n* potrebných $O(n^2)$ kôl. ♣

Nasledujúce 3 algoritmy sú nekonštruktívne. Sú založené na tom, že pravdepodobnostnými metódami je ukázaná existencia akéhosi kombinatorického objektu s istými vlastnosťami. Za predpokladu, že tento objekt je pre daný parameter *n* známy všetkým vrcholom siete, realizujú tieto algoritmy rádiový broadcasting v uvedených časoch. Ak je kombinatorický objekt známy len zdrojovému vrcholu, môže byť ako súčasť posielaných správ doručený vrcholom pred tým, než budú mať podľa neho vysielať. Žiaľ, nie sú známe algoritmy, ktoré umožňujú používané kombinatorické objekty efektívne generovať.

Tvrdenie 3.2.2 [CGR00]**:** *Nech* $G = (V, E)$ *je orientovaný graf dosiahnuteľnosti rádiovej siete a informácia v každom uzle siete je obmedzená identifikátor vrcholu a počet vrcholov siete n. Potom existuje algoritmus, ktorý realizuje rádiový broadcasting (RB) počas* $O(n \log^2 n)$ *kôl.*

Idea dôkazu: Algoritmus je založený na koncepte výberových množín (selective families).

Definícia: *Nech množiny* $S, X \subseteq \{1, \ldots, n\}$. *Potom hovoríme, že*

- *množina S zasahuje množinu X práve vtedy, ak* $|S \cap X| = 1$.

- *množina S sa vyhýba množine X práve vtedy, ak* $S \cap X = \emptyset$.

Pre dané prirodzené číslo w hovoríme, že množina množín $\overline{S}$ *je w-selektor, ak pre ľubovoľné množiny* $X, Y \subseteq \{1, \ldots, n\}$ *s vlastnosťou* $w/2 \leq |X| \leq w$ *a* $|Y| \leq w$ *platí, že existuje množina* $S \in \overline{S}$, *ktorá zasahuje X a vyhýba sa Y.*

Lema: *Pre každé n a každé prirodzené číslo* $w \leq n$ *existuje w-selektor* $\overline{S}$ *taký, že* $|\overline{S}| \in O(w \log n)$.

Nech pre každé $j = 0, \ldots, \log n$ je $\overline{S}_j = (S_{j,0}, S_{j,1}, \ldots, S_{j,m_j - 1})$ je 2^j-selektor pozostávajúci z $m_j \in O(2^j \log n)$ množín. Predchádzajúca lema garantuje existenciu týchto selektorov.

Samotný algoritmus pracuje vo fázach, pričom každá fáza má $\log n + 1$ kôl. V *i*-tom kole fázy *k* vysielajú tie vrcholy, ktoré prijali zdrojovú správu a ich identifikátor $id \in S_{i, k \bmod m_j}$.

Po $O(n \log n)$ fázach všetky vrcholy príjmu zdrojovú správu. ♣

Tvrdenie 3.2.3 [KPa03]**:** *Nech* $G = (V, E)$ *je orientovaný graf dosiahnuteľnosti rádiovej siete, kde informácia v každom uzle siete je obmedzená identifikátor vrcholu a počet vrcholov siete n. Potom existuje algoritmus, ktorý realizuje rádiový broadcasting (RB) za* $O(n \log n \log D)$ *kôl.*

Tvrdenie 3.2.4 [CR03]**:** *Nech* $G = (V, E)$ *je orientovaný graf dosiahnuteľnosti rádiovej siete kde, informácia v každom uzle siete je obmedzená identifikátor vrcholu a počet vrcholov siete n. Potom existuje algoritmus, ktorý realizuje rádiový broadcasting (RB) za* $O(n \log^2 D)$ *kôl.*

Idea dôkazu: Algoritmus vychádza z myšlienky randomizovaného algoritmu, ktorý autori navrhli. Deterministická verzia kombinuje vysielanie podľa selektorov na základe postupnosti, ktorá spĺňa predpoklady randomizovaného vysielania. ♣

V nasledujúcich 3 tvrdeniach uvedených bez dôkazov si zhrnieme známe dolné odhady týkajúce sa distribuovaného rádiového broadcastingu v rádiových sieťach bez detekcie kolízie. Poznamenajme, že pre distribuovaný rádiový broadcasting platia aj dolné odhady pre centralizovaný broadcasting.

Tvrdenie 3.2.5 [CGGPR00]: *Nech P je deterministický distribuovaný protokol realizujúci RB a využívajúci spontánne vysielania, kde vrcholy siete sa správajú na základe obsahu prijatých správ, čísla aktuálneho kola a akýchkoľvek globálnych parametrov siete. Potom pre každé n a pre každé* $D < 2n/3$ *existuje orientovaný planárny graf G o n vrcholoch s excentricitou zdroja D taký, že P vykoná broadcasting za* $\Omega(D \log n)$ *kôl.*

Tvrdenie 3.2.6 [CMS01]**:** *Nech P je deterministický distribuovaný protokol realizujúci RB, kde vrcholy siete sa správajú na základe počtu vrcholov n v sieti, maximálnej excentricity D všetkých možných zdrojových vrcholov a obsahu prijatých správ. Potom pre každé n a pre každé* $D \le n/6$ *existuje orientovaný graf* G^P *o n vrcholoch s maximálnou excentricitou D taký, že P vykoná broadcasting za* $\Omega(n \log D)$ *kôl.*

Tvrdenie 3.2.7 [KPb03]: *Nech P je deterministický distribuovaný protokol realizujúci RB, kde vrcholy siete sa správajú na základe počtu vrcholov n v sieti a obsahu prijatých správ. Potom pre každé n a pre každé* $D \le n$ *existuje neorientovaný planárny graf* G_P *o n vrcholoch s excentricitou zdroja D taký, že P potrebuje na uskutočnenie broadcastingu* $\Omega(n \frac{\log n}{\log(n/D)})$ *kôl.*

Symetrické rádiové siete

Pozrime sa teraz na výsledky v oblasti broadcastingu vo všeobecných symetrických sieťach – čiže v rádiových sieťach, ktorých graf dosiahnuteľnosti je neorientovaný. Poznamenajme, že všetky dolné odhady času rádiového broadcastingu dokázané pre symetrické siete platia aj pre všeobecné siete a naopak, všetky horné odhady dokázané pre všeobecné (nesymetrické) siete platia aj pre symetrické siete.

V práci [CGGPR00] autori skonštruovali algoritmus realizujúci distribuovaný rádiový broadcasting v čase $O(n)$. Algoritmus však silne využíva spontánne vysielania – t.j. všetky vrcholy v sieti poznajú začiatok vykonávania algoritmu a môžu vysielať skôr, než príjmu prvú správu ako dôsledok vykonávania algoritmu.

Tvrdenie 3.2.8 [CGGPR00]**:** *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý realizuje rádiový broadcasting s využitím spontánnych vysielaní počas* $O(n)$ *kôl.*

Idea dôkazu (Algoritmus EXPLORE-EXPAND): Každý vrchol si počas algoritmu buduje zoznam svojich susedov a spolu s informáciou, či už poznajú zdrojovú správu. Algoritmus pracuje vo fázach, pričom fáza *k* sa skladá z 7.2^{k-1} kôl a je rozdelená na 3 časti. Časť A trvá 2^{k-1} kôl, časť B 2^k kôl a časť C 2^{k+1} kôl. Cieľom fázy *k* je komponente súvislosti grafu $G_k = G(V_k)$, kde $V_k = \{v \in V \mid id(v) \le 2^k\}$, ktorá obsahuje zdroj, skonštruovať orientovaný Eulerovský ťah C_k DFS kostry grafu G_k. Tento ťah slúži ako „rýchla" komunikačná linka medzi zdrojom a ostatnými vrcholmi grafu G_k. Každý vrchol si pamätá svoje poradové číslo v tomto ťahu C_k, pričom poradové číslo zdroja je 0. Ak zdroj neleží v G_k, ťah C_k sa nekonštruuje.

V časti A postupne vysielajú správu so svojim identifikátorom vrcholy s identifikátorom $2^{k-1}+1,\ldots,2^{k-1}+2^{k-1}$ (2^{k-1} vrcholov počas 2^{k-1} kôl). Ich susedia, ktorí v tejto časti prijímajú správu si aktualizujú zoznam susedov. V časti B v *j*-tom kole vysiela kontaktnú správu ten vrchol, ktorý má v ťahu C_{k-1} poradové číslo j za predpokladu, že v časti A alebo predchádzajúcom kole časti B prijal správu. Poznamenajme, že jeden vrchol môže mať v ťahu C_{k-1} viacero poradových čísel. Cieľom časti B je oznámiť zdroju, že došlo k rozšíreniu komponentu súvislosti o nové vrcholy. Všimnime si tiež, že po skončení častí A a B, každý vrchol v grafe G_k pozná identifikátory všetkých svojich susedov v grafe G_k. V časti C sa DFS prehľadávaním grafu G_k pomocou tokenu obsahujúceho: zdrojovú správu, dĺžku cesty prejdenej tokenom (poradové číslo vrcholu v rámci ťahu) a identifikátor (susedného) vrcholu, ktorému je token určený. Časť C fázy *k* skonštruuje ťah C_k. ♣

Mohlo by sa zdať, že algoritmus z predchádzajúcej vety realizuje ARB. Všimnime si, že môže nastať prípad, kedy susednými vrcholmi zdroja sú vrcholy s veľmi vysokými identifikátormi. Vrcholom siete ale podľa použitého modelu nie je známe ohraničenie počtu vrcholov. Zdroj teda nedokáže rozlíšiť či to, že od začiatku algoritmu do aktuálneho kola neprijal žiadnu správu, je dôsledkom toho, že je jediným vrchol v sieti, alebo toho, že ešte nenastala taká fáza *k*, v ktorej by mali vysielať jeho susedné vrcholy. Na základe podobných úvah je v práci [CGGPR00] formálne dokázaná nasledujúca vetu o nemožnosti ARB už v prípade symetrických sietí:

Tvrdenie 3.2.9 [CGGPR00]: *Nech P je ľubovoľný deterministický distribuovaný protokol realizujúci ARB s možnosťou spontánnych vysielaní, kde vrcholy poznajú len svoj identifikátor. Potom existuje symetrický planárny graf s excentricitou zdroja 2, na ktorom P nepracuje korektne.*

V prípade, že spontánne vysielania nepoužívame, je situácia z časového hľadiska výrazne horšia. Kowalski a Pelc v práci [KPb03] skonštruovali algoritmus realizujúci RB počas $O(n\log n)$ kôl. Použitá je v ňom zaujímavá technika simulovania detekcie kolízie.

Tvrdenie 3.2.10 [KPb03]: *Nech $G=(V,E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý realizuje rádiový broadcasting počas $O(n\log n)$ kôl.*

Idea dôkazu (Algoritmus Select-and-Send): Predpokladajme, že vrchol v prijal zdrojovú správu. Nech $A\subset\Gamma(v)$ je vlastná podmnožina množiny susedných vrcholov a nech $w\notin A$ je význačný sused vrcholu *v*. Cieľom je rozlíšiť, či množina *A* má 0, 1 alebo viac ako jeden prvok. Na tento účel môžeme použiť procedúru *Echo* pracujúcu v 2 kolách:

Procedure *Echo(w, A)*

1. *kolo:* Každý vrchol v množine A vysiela svoj identifikátor
2. *kolo:* Každý vrchol v množine $A \cup \{w\}$ vysiela svoj identifikátor

Počas vykonávania procedúry *Echo* môžu nastať 3 prípady:

1. vrchol *v* prijme správu v 1. kole a neprijme žiadnu správu v 2. kole: V tomto prípade vrchol *v* vie, že $|A| = 1$.
2. vrchol *v* neprijme v 1. kole žiadnu správu a v 2. kole prijme správu (od vrcholu *w*). V tomto prípade vrchol *v* vie, že $|A| = 0$.
3. vrchol *v* neprijme v 1. ani v 2. kole žiadnu správu. Vrchol *v* tak vie, že $|A| \geq 2$.

Predpokladajme, že vrchol *v* pozná jedného zo svojich susedov *w*. Nech *S* je množina susedov vrcholu *v* rôznych od *w*. Nech ďalej vrchol *v* pozná horné ohraničenie *m* identifikátorov vrcholov v množine *S*, pričom *m* je mocninou 2. Potom pomocou procedúry *Echo* môžeme vybrať jeden vrchol z množiny S v čase $O(\log m)$ algoritmom *Binary-Selection*. Algoritmus *Binary-Selection* pracuje vo fázach, z ktorých každá ma 3 kolá. V prvom kole každej fázy vysiela vrchol *v* rozsah identifikátorov *R*, pričom v prvej fáze je tento rozsah $R := \{1, \ldots, m/2\}$. V nasledujúcich 2 kolách fázy potom vykoná procedúru $Echo(w, R \cap S)$. Nech rozsah v danej fáze je $R = \{x, \ldots, y\}$. Podľa výsledku vykonania procedúry *Echo* potom:

Ak $|R \cap S| = 1$, môže vrchol *v* vybrať jedného zo susedov.
Ak $|R \cap S| = 0$, tak nastaví pre ďalšiu fázu $R := \{y+1, \ldots, y + (y - x + 1)/2\}$.
Ak $|R \cap S| \geq 2$, tak nastaví pre ďalšiu fázu $R := \{x, \ldots, (y - x - 1)/2\}$.

Nakoniec môžeme procedúru *Echo* spolu s algoritmom *Binary-Selection* použiť na samotný broadcastovací algoritmus *Select-and-Send*. Algoritmus postupne DFS prehľadávaním prechádza graf, pričom v každom okamihu drží token (právo vysielať) jeden z vrcholov grafu. Na začiatku drží token zdrojový vrchol. Ten si najprv v prvej časti algoritmu vyberie jedného zo susedov následovne: Zdroj v 1. kole vysiela povel na začatie výberu jedného zo susedov. Vrcholy, ktoré túto správu prijali, potom vysielajú tak, že v kole $2i$ vrchol s identifikátorom i vysiela správu so svojim identifikátorom. Ak zdroj v kole $2i$ prijíma takúto správu, tak v kole $2i+1$ vysiela správu s príkazom pre susedov na ukončenie výberu. Po prijatí tejto správy susedia zdroja prestávajú vykonávať výber a zdroj pozná identifikátor jedného so svojich susedov. Následne začína druhá časť algoritmu – vysielanie tokenu. Zdroj vysiela token zvolenému vrcholu s identifikátorom *i*. Token obsahuje okrem identifikátora odosielateľa a príjemcu aj zdrojovú správu. Pre každý vrchol okrem zdrojového nech $parent(v)$ je identifikátor vrcholu, od ktorého prijal token po prvý krát. Každý vrchol po prijatí tokenu pomocou procedúry *Echo* skontroluje, či existuje sused, ktorý ešte neprijal token. Ak taký sused neexistuje, vráti ho vrcholu $parent(v)$. V opačnom prípade sa pokúsi vybrať jedného zo susedov, ktorý ešte neprijal token. Nech S je množina susedov vrcholu, ktorí ešte neprijali token. Vrchol pomocou procedúry *Echo* najprv nájde ohraničenie identifikátora aspoň jedného vrcholu z množiny S. Postupne pre $k \geq 0$ spúšťa procedúru $Echo(parent(v), S \cap \{0,\ldots,2^k\})$, až kým nenájde k také, že $S \cap \{0,\ldots,2^k\} \neq \emptyset$. Potom algoritmom *Binary-Selection* vyberie jeden susedný vrchol z množiny S, ktorému pošle token. Poznamenajme, že pri procedúre *Echo* ako pomocný vrchol používame vždy vrchol $parent(v)$ v prípade nezdrojového vrcholu a vrchol vybraný v prvej časti algoritmu v prípade zdrojového vrcholu. Algoritmus končí, keď zdrojový vrchol nemá komu poslať token. Počet kôl potrebných na odoslanie tokenu vrcholu, ktorý ešte neprijal žiaden token je $O(\log n)$. Celkovo je tak zložitosť tohto algoritmu $O(n \log n)$ kôl. ♣

3.3 Distribuovaný broadcasting s detekciou kolízie

V tejto podkapitole sa pozrieme na distribuovaný rádiový broadcasting v modely s detekciou kolízie. Kým v symetrických sieťach bez detekcie kolízie ARB nie je možný, v modely s detekciou kolízie je možný už aj v prípade silne súvislých grafov. V symetrických sieťach s možnosťou spontánnych vysielaní je možné dosiahnuť ARB počas $O(n)$ kôl. Zároveň detekcia kolízie so sebou prináša možnosť zakódovania informácií do kolízie (viď tvrdenie 3.3.1). V tejto podkapitole budeme k*ontaktnou správou* nazývať jednobitovú správu, resp. ľubovoľnú správu, ktorej obsah hovorí, že ide o takýto typ správy.

Tvrdenie 3.3.1 [CGGPR00]: *Nech $G = (V, E)$ je ľubovoľný orientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny algoritmus, ktorý realizuje rádiový broadcasting zdrojovej správy σ počas $O(|\sigma| D)$ kôl, kde D je excentricita zdroja a $|\sigma|$ je počet bitov zdrojovej správy.*

Idea dôkazu (Algoritmus ENCODED-BROADCAST): Nech $(a_1, \ldots, a_r)$ je binárna reprezentácia zdrojovej správy σ. Algoritmus pracuje vo fázach. Prvá fáza sa skladá z 1 kola, každá ďalšia fáza sa skladá z $2r + 4$ kôl. V každej fáze sú niektoré vrcholy aktívne a všetky ostatné sú pasívne. V prvej fáze je aktívny len zdroj a vysiela zdrojovú správu σ počas 1. kola tejto fázy. Vo fáze k sú aktívne tie vrcholy, ktoré počas predchádzajúcej fázy $k-1$ prijali v nejakom kole správu alebo detekovali kolíziu, a v žiadnej fáze doteraz ešte neboli aktívne. Ako bude vidieť neskôr, sú to tie vrcholy, ktoré počas fázy $k-1$ dekódovali zdrojovú správu a teda ju na začiatku fázy k už poznajú. Invariantom algoritmu je, že všetky aktívne vrcholy poznajú zdrojovú správu. Fáza k je rozdelená do $r+2$ dvojkolových segmentov

$b_0, b_1, \ldots, b_r, b_{r+1}$. Aktívne vrcholy v oboch kolách segmentov b_0 a b_{r+1} vysielajú kontaktnú správu. Vysielanie v segmente b_i pre $1 \le i \le r$ závisí od hodnoty a_i. Ak $a_i = 0$, tak aktívne vrcholy pracujú v oboch kolách segmentu b_i ako prijímače, t.j. nevysielajú žiadnu správu. Ak $a_i = 1$, tak v prvom kole segmentu b_i vystupujú aktívne vrcholy ako prijímače (nevysielajú správu) a v druhom kole vysielajú kontaktnú správu. Vrcholy, ktoré sú aktívne vo fáze $k+1$ poznajú na začiatku tejto fázy zdrojovú správu, pretože ju vedia dekódovať z vysielania vo fáze k. Všetky aktívne vrcholy totiž vysielajú rovnako. Teda buď v danom kole všetky vrcholy vystupujúce ako prijímače neprijímajú žiadnu správu, alebo všetky rozpoznajú vyslanie kontaktnej správy (buď ako prijatie kontaktnej správy alebo ako detekciu vzniku kolízie). Cieľom segmentov b_0 a b_{r+1} je ohraničiť kolá obsahujúce vysielanie správy. Rozdelením kôl vysielania na dvojice zodpovedajúce segmentom a rozpoznaním toho, či aktívne vrcholy vysielali alebo nevysielali, vedia vrcholy aktívne vo fáze $k+1$ priamočiaro dekódovať obsah zdrojovej správy. ♣

Tvrdenie 3.3.2 [CGGPR00]: *Nech $G = (V, E)$ je silne súvislý graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý realizuje ARB s použitím spontánnych vysielaní počas $O(nD)$ kôl.*

Idea dôkazu (Algoritmus BOUND-BROADCAST-CHECK): Algoritmus pozostáva z 3 častí nasledujúcich za sebou. Cieľom prvej časti je určiť horné ohraničenie počtu vrcholov siete. Druhá časť má za cieľ, aby vrcholy v sieti spoznali svoju vzdialenosť od zdroja (príslušnosť k vrstve). Tretia časť algoritmu realizuje súčasne samotný broadcasting zdrojovej správy a výpočet excentricity zdroja tak, aby táto hodnota bola známa všetkým vrcholom siete. Znalosť excentricity umožňuje zastavenie vykonávania algoritmu, čiže ARB.

Prvá časť algoritmu pracuje vo fázach, pričom fáza k trvá $2^k + 1$ kôl. Hovoríme, že vrchol je aktívny vo fáze k, ak aspoň v jednom kole fázy k vystupoval ako vysielač. V 1. fáze sú aktívne všetky vrcholy. Ako uvidíme neskôr, v každej fáze budú aktívne buď všetky vrcholy alebo žiaden. V 1. kole fázy vysielajú kontaktnú správu tie vrcholy, ktorých identifikátor je väčší než 2^k. Ostatné vrcholy vystupujú ako prijímače. V *i*-tom kole ($i \ge 1$) fázy vysielajú kontaktnú správu tie vrcholy, ktoré po prvý krát v rámci aktuálnej fázy v kole $i-1$ prijali kontaktnú správu alebo detekovali kolíziu. Nech k_0 je prvá fáza, v ktorej nie je aktívny žiaden vrchol. Potom prvá časť algoritmu končí a všetky vrcholy vedia, že pre najväčší identifikátor *Z* v sieti platí $2^{k_0 - 1} < Z \le 2^{k_0}$ pričom $Z = O(n)$.

V druhej časti algoritmu, ktorá trvá 2^{k_0} kôl, sa všetky vrcholy dozvedia svoju vzdialenosť od zdroja. V 1. kole vysiela zdroj kontaktnú správu. V kole $i > 1$ vysielajú kontaktnú správu tie vrcholy, ktoré v kole $i-1$ prijali kontaktnú správu alebo detekovali kolíziu po prvý krát v 2. časti algoritmu. Na základe toho, v ktorom kole vysielajú kontaktnú správu, poznajú svoju vzdialenosť od zdroja.

Tretia časť algoritmu prebieha vo fázach, ktoré sú rozdelené na dve po sebe nasledujúce podfázy po 2^{k_0} kôl. V prvej podfáze v kole *i* vysiela zdrojovú správu vrchol s identifikátorom *i*, ak túto zdrojovú správu pozná. Druhá podfáza funguje ako jedna z fáz v prvej časti algoritmu s tým rozdielom, že v prvom kole druhej podfázy fázy k vysielajú kontaktnú správu tie vrcholy, ktorých vzdialenosť od zdroja je väčšia než k. Ak po skončení podfázy nie je žiaden vrchol aktívny (neprijala sa kontaktná správu ani nedetekovala kolízia), znamená to úspešné vykonanie broadcastingu. ♣

Tvrdenie 3.3.3 [CGGPR00]: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý realizuje ARB s použitím spontánnych vysielaní počas* $O(n)$ *kôl.*

Idea dôkazu (Algoritmus EXPLORE-EXPAND-CHECK): Algoritmus je modifikáciou algoritmu *EXPLORE-EXPAND* z tvrdenia 3.2.8. Na zistenie ukončenia broadcastingu je algoritmus doplnený v každej fáze o jedno kolo vysielania. V ňom vrcholy s identifikátorom väčším než 2^k vysielajú kontaktnú správu. Ak v tomto kole vrcholy s identifikátorom nanajvýš 2^k vystupujúce ako prijímače príjmu kontaktnú správu alebo detekujú kolíziu, tak prejdú do stavu *varovaný*. Zároveň je upravená aj časť C algoritmu tak, že ak token príde do varovaného vrcholu, zaznamená sa to do tokenu. Ak zdrojový vrchol z prijatého tokenu zistí, že počas svojej cesty neprešiel cez varovaný vrchol, vie, že broadcasting je ukončený. ♣

Algoritmus z tvrdenia 3.3.3 realizuje ARB v čase $O(n)$ v rádiových sieťach s detekciou kolízie, ktorých graf dosiahnuteľnosti je symetrický. Algoritmus však využíva spontánne vysielania. V modely bez spontánnych vysielaní môžeme použiť algoritmus *ENCODED-BROADCAST* z tvrdenia 3.3.1, ktorý realizuje RB počas $O(|\sigma|D)$, kde $|\sigma|$ je počet bitov správy σ. V prípade, že zdrojové správy sú dlhé alebo prístup k bitovému zápisu správy nemáme, tento algoritmus je nevýhodný, resp neefektívny. V nasledujúcej časti ukážeme algoritmus, ktorý realizuje ARB počas $O(n + D \log D)$ kôl bez využitia spontánnych vysielaní v rádiových sieťach s detekciou kolízie, ktorých graf dosiahnuteľnosti je symetrický, a jediná informácia vo vrchole je jeho identifikátor.

Nasledujúce algoritmy vychádzajú z myšlienky algoritmu *ENCODED-BROADCAST* (tvrdenie 3.3.1), kde je schopnosť detekcie kolízie využitá na dekódovanie správy zakódovanej do série vysielaní kontaktných správ.

Distribuované priradenie čísla vrstvy vrcholu

Tvrdenie 3.3.4: *Nech $G = (V, E)$ je graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý pre každý vrchol v sieti určí jeho vzdialenosť od zdroja, t.j. jeho zaradenie do vrstvy.*

Dôkaz (Algoritmus ENCODED-LAYERING): Algoritmus bude pracovať vo fázach. Vo fáze i ($i \geq 0$) budú aktívne všetky vrcholy vo vrstve i. V algoritme udržiavame invariant, že všetky vrcholy vo vrstve i vstúpia do fázy i v rovnakom kole a pred vstupom do prvého kola tejto fázy poznajú svoju príslušnosť do vrstvy i. Všetky ostatné vrcholy budú neaktívne a vystupujú ako prijímače. Počas každej z fáz vrcholy vyšlú vrcholom nasledujúcej vrstvy poradové číslo vrstvy, do ktorej vrcholy tejto nasledujúcej vrstvy patria – teda vo fáze i sa vysiela číslo $i+1$. Toto číslo vyšlú zakódované do série vysielaní. Nech $(i_1, \ldots, i_r)$ je binárny zápis čísla $i+1 \geq 1$. Potom fáza i bude pozostávať z $2r+4$ kôl, ktoré sú rozdelené do $r+2$ dvojkolových segmentov $b_0, b_1, \ldots, b_r, b_{r+1}$. V oboch kolách segmentu b_0 vystupujú všetky aktívne vrcholy (vrcholy vrstvy i) ako vysielače a vysielajú kontaktnú správu. Zoberme segment b_j, $1 \leq j \leq r$ prislúchajúci hodnote i_j. Ak $i_j = 0$, tak aktívne vrcholy v oboch kolách segmentu budú vystupovať ako prijímače – t.j. nevysielajú žiadnu správu. Ak $i_j = 1$, tak aktívne vrcholy budú v prvom kole segmentu vystupovať ako prijímače a v druhom kole segmentu ako vysielače, pričom vyšlú kontaktnú správu. Nakoniec v segmente b_{r+1} budú aktívne vrcholy vystupovať v oboch kolách segmentu ako vysielače a vyšlú kontaktnú správu. Poznamenajme, že vzhľadom na udržiavaný invariant, všetky aktívne vrcholy vysielajú vo všetkých kolách rovnako. Teda v kole, v ktorom majú aktívne vrcholy vysielať, všetky ich susedné vrcholy vystupujúce ako prijímače príjmu kontaktnú správu alebo detekujú kolíziu. Spomedzi neaktívnych vrcholov budú vyslanú správu dekódovať iba tie vrcholy, ktoré svoju príslušnosť do vrstvy ešte nemajú určenú, t.j. iba vrcholy vrstvy $i+1$. Vysielanie počas segmentu b_0 má za následok „zobudenie“ vrcholov vrstvy $i+1$ a ich pripravenie na príjem správy zakódovanej do série vysielaní. Zároveň vysielanie v segmente b_{r+1} informuje o ukončení vysielania správy. Vysielanie v oboch kolách týchto segmentov (b_0 a b_{r+1}) zaručuje, že toto vysielanie nebude zamenené za vysielanie obsahu správy počas segmentov $b_1, \ldots, b_r$. V každej fáze je vďaka tomu možné vysielať správu inej dĺžky.

Dekódovanie vyslaného čísla $i+1$ bude vo vrcholoch vrstvy $i+1$ prebiehať segmentoch ohraničených segmentmi b_0 a b_{r+1} - t.j. v segmentoch, kde vrcholy počas prvého kola neprijímajú žiadnu správu ani nedetekujú kolíziu. Ak je v druhom kole vysielania segmentu b_j prijatá kontaktná správa alebo je detekovaná kolízia, tak *j*-ty bit vysielanej zakódovanej správy je *1*, inak je tento bit rovný *0*. Vrcholy po prijatí správy a dekódovaní jej obsahu si toto číslo uložia ako svoju príslušnosť k vrstve a stanú sa aktívnymi v nasledujúcej fáze. Aktívne vrcholy vo fáze i sa stanú neaktívnymi.

Uvedený algoritmus vykoná distribuované priradenie čísla vrstvy vrcholu na nanajvýš $O(D \log D)$ kôl, keďže najdlhšia vysielaná správa má dĺžku $O(\log D)$ a algoritmus pozostáva z $D+1$ fáz. ♣

Poznamenajme ďalej, že uvedený algoritmus je možné rozšíriť tak, že umožní relatívnu synchronizáciu čísla aktuálneho kola v jednotlivých vrcholoch vzhľadom na začiatok algoritmu v zdrojovom vrchole. Algoritmus je treba upraviť tak, že každý vrchol, si na základe svojej príslušnosti k vrstve vypočíta a nastaví číslo aktuálneho kola. Konkrétne tento výpočet je založený na fakte, že vrchol vo vrstve i ukončuje príjem svojho zaradenia do vrstvy po $\sum_{r=0}^{i-1} 2\lfloor \log(r+1) \rfloor + 6$ kolách algoritmu. Ak teda algoritmus začal v kole 1, tak kolo $\sum_{r=0}^{i-1} 2\lfloor \log(r+1) \rfloor + 6$ je posledným kolom fázy $i-1$.

Distribuované určenie odhadu najväčšej hodnoty v sieti

Tvrdenie 3.3.5: *Nech $G=(V,E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Ďalej nech každému vrcholu $v \in V$ v sieti je priradené celé číslo $Value(v) \geq 0$. Potom existuje konštruktívny distribuovaný algoritmus, ktorý počas $O(D(\log D + \log Z))$ kôl vypočíta v zdroji siete také ohraničenie Z, že pre najväčšiu hodnotu priradenú vrcholu v sieti $Value_{MAX} = \max\{Value(v) \mid v \in V\}$ platí, že $Z/2 < Value_{MAX} \leq Z$.*

Dôkaz (Algoritmus ENCODED-FIND-BOUND): Predpokladajme najprv, že pracujeme v modely, kde informácia v každom vrchole je obmedzená na identifikátor vrcholu a jeho zaradenie do vrstvy. Neskôr ukážeme ako predpoklad znalosti zaradenia vrcholu do vrstvy odstrániť.

Základnou ideou algoritmu je akoby prehľadaním do šírky nájsť ohraničenie hodnoty $Value_{MAX}$ - najväčšej hodnoty priradenej nejakému vrcholu siete. Na začiatku zdrojový vrchol vysiela povel na začatie algoritmu, ktorý sa prenesie po vrstvách do všetkých vrcholov siete. Každý vrchol čaká, kým si jeho susedné vrcholy v nasledujúcej vrstve neurčia vlastné odhady tohto ohraničenia. Potom vrchol zozbiera odhady susedných vrcholov z nasledujúcej vrstvy takým spôsobom, že i keď nezíska jednotlivé odhady susedov presne, získa nejakú hodnotu najväčšieho odhadu, aká sa vyskytuje u susedných vrcholov. Túto hodnotu porovná so svojou priradenou hodnotou. Získa tak vlastný odhad najväčšej hodnoty priradenej nejakému vrcholu v sieti. Následne je vrchol pripravený oznámiť svoj odhad na požiadanie susedným vrcholom z predchádzajúcej vrstvy. Takto sa odhady postupne dostanú spätne po vrstvách až k zdrojovému vrcholu. Odhad najväčšej hodnoty priradenej vrcholu v sieti budeme určovať ako odhad v tvare 2^i.

Algoritmus bude pracovať vo fázach. V každom kole algoritmu môže byť vrchol v stave *neaktívny, oznamujúci, čakajúci, vediaci, prijímajúci* alebo *vysielajúci*. Na začiatku je zdrojový vrchol v stave *oznamujúci* a všetky ostatné vrcholy sú v stave *neaktívny*. Každá fáza je rozdelená do 3 podfáz (číslovaných *0, 1* a *2*), z ktorých každá sa skladá zo 4 dvojkolových segmentov – t.j. každá fáza pozostáva celkovo z 24 kôl. V druhom kole segmentu sa vysiela správa iba v niektorých prípadoch počas prvého segmentu.

Vrcholy v stave *oznamujúci* oznamujú vrcholom v nasledujúcej vrstve vykonávanie algoritmu. V stave *čakajúci* sú tie vrcholy, ktoré čakajú, kým všetky ich susedné vrcholy v nasledujúcej vrstve nepoznajú svoj odhad najväčšej hodnoty priradenej nejakému vrcholu siete. Vrcholy v stave *vediaci* už majú svoj odhad určený. Vrcholy v stave *vysielajúci* vysielajú svoj odhad vrcholom predchádzajúcej vrstvy, ktoré sú v stave *prijímajúci*.

Pre každú podfázu *j* nazvime *vrstvami podfázy j* množinu vrcholov $S_j = \bigcup_{i \equiv j \pmod 3} L_i$. Ďalej pre podfázu *j* definujme množiny $P_j^0 := S_j$, $P_j^{+1} := S_{(j+1) \bmod 3}$ a $P_j^{-1} := S_{(j-1) \bmod 3}$.

V *j*-tej podfáze $0 \le j \le 2$ prebieha vysielanie následovne:

<u>V prvom segmente</u> vysielajú v oboch kolách kontaktnú správu všetky vrcholy v stave *oznamujúci* z vrstiev aktuálnej podfázy, t.j. z množiny P_j^0. Po odvysielaní správ prejdú tieto vrcholy do stavu *čakajúci*. Vrcholy v stave *neaktívny*, ktoré v nejakých dvoch za sebou nasledujúcich kolách prijímajú kontaktnú správu alebo detekujú kolíziu, prejdú do stavu *oznamujúci*. Vysielanie v podfáze je totiž naplánované tak, že jediné kolá, v ktorých môže nastať takáto situácia, sú kolá, v ktorých vysielajú vrcholy predchádzajúcej vrstvy počas prvého „oznamovacieho" segmentu príslušnej podfázy. Podľa vrstvy, do ktorej vrcholy patria, si vedia pri zmene stavu z *neaktívny* na *oznamujúci* určiť práve prebiehajúce kolo v rámci aktuálnej fázy. Konkrétne pre vrchol z vrstvy *i* táto zmena stavu nastáva po 2. kole prvého segmentu podfázy *j*, kde $i - 1 \equiv j \pmod 3$. Cieľom tejto časti je „aktivovať" vrcholy siete, aby začali participovať na vykonávanom algoritme a zároveň sa zosynchronizovali v rámci práve prebiehajúcej fázy. Poznamenajme, že aj keď sa toto kolo vykonáva v každej podfáze každej fázy, k reálnemu vysielaniu dôjde len počas prvých $D+1$ podfáz.

<u>V druhom segmente</u> podfázy vysielajú v prvom kole kontaktnú správu tie vrcholy v stave *oznamujúci, čakajúci* alebo *prijímajúci*, ktoré ležia vo vrstvách nasledujúcich vrstvy aktuálnej podfázy, t.j. vrcholy z množiny P_j^{+1}. Ak vrchol z P_j^0, t.j. vo vrstve aktuálnej podfázy, ktorý je v stave *čakajúci*, počas tohto kola neprijme kontaktnú správu ani nedetekuje kolíziu, prejde do stavu *prijímajúci*. Všimnime si, že vrchol z P_j^0 mení svoj stav, ak všetci jeho susedia v nasledujúcej vrstve (t.j. z P_j^{+1}) sú v stave *vysielajúci* alebo *vediaci*. To, ako uvidíme neskôr, znamená, že všetky susedné vrcholy už poznajú svoj odhad najväčšej hodnoty priradenej nejakému vrcholu v sieti. V druhom kole tohto segmentu vystupujú vrcholy z vrstiev aktuálnej podfázy ako prijímače, t.j. nevysiela sa žiadna správa.

V treťom segmente vysielajú kontaktnú správu v prvom kole tie vrcholy z P_j^0, ktoré v predchádzajúcom segmente prešli do stavu *prijímajúci*. Týmto kolom oznamujú vrcholy, ktoré zistili, že už všetky ich susedné vrcholy v nasledujúcej vrstve poznajú svoj odhad, aby im tieto susedné vrcholy svoje odhady zaslali. Vrcholy z P_j^{+1}, t.j. vo vrstvách nasledujúcich vrstvy aktuálnej podfázy, ktoré v tomto kole prijímajú kontaktnú správu alebo detekujú kolíziu a sú v stave *vediaci*, prechádzajú do stavu *vysielajúci*. Rovnako aj v tomto segmente sa v druhom kole nevysiela žiadna správa.

V štvrtom segmente vysielajú v prvom kole kontaktnú správu vrcholy v stave *vysielajúci* z množiny P_j^{+1}, t.j. z vrstiev nasledujúcich vrstvy aktuálnej podfázy. Podobne ako v predchádzajúcich segmentoch sa v druhom kole nevysiela. Ak 2^r je v danom vrchole odhadovaná hodnota najväčšej priradenej hodnoty v sieti, tak tento vrchol bude celkovo, od zmeny svojho stavu na stav *vysielajúci*, vysielať v prvom kole štvrtého segmentu príslušnej podfázy v r nasledujúcich fázach kontaktnú správu. Po poslednom odvysielaní kontaktnej správy prechádza vrchol do stavu *neaktívny*. Vrcholy v stave *prijímajúci* vo vrstvách aktuálnej podfázy (P_j^0) počítajú, v koľkých fázach za sebou v prvom kole štvrtého segmentu prijímajú kontaktnú správu alebo detekujú kolíziu. Ak v niektorej z fáz neprijmú žiadnu správu v štvrtom segmente, znamená to, že všetky odhady susedov už boli odvysielané. Nech r je počet za sebou idúcich fáz, v ktorých vrchol v stave *prijímajúci* prijal kontaktnú správu alebo detekoval kolíziu v štvrtom segmente príslušnej podfázy. Potom vrchol prejde do stavu *vediaci* a nastaví ako odhadovanú hodnotu ohraničenia najväčšej priradenej hodnoty v sieti na číslo $\max(2^r, 2^{\lceil \log Value(v) \rceil})$, kde $Value(v)$ je jemu priradená hodnota.

Rozdelenie vysielanie do podfáz podľa príslušnosti k vrstve zaručuje, že počas vysielania nedôjde k neželanej interferencii. Zároveň, vďaka znalosti príslušnosti k vrstve, na vysielané správy v danej podfáze môžu reagovať len tie vrcholy, pre ktoré je toto vysielanie určené. V rámci podfázy j chceme, aby navzájom komunikovali iba vrcholy z množín P_j^0 a P_j^{+1}. Všimnime si, že vrcholy, ktoré v nasledujúcej vrstve nemajú susedov (a teda od nich neprijmú žiadne vysielanie) sa podľa algoritmu správajú tak, ako keby im z ďalšej vrstvy prišiel odhad ohraničenia najväčšej priradenej hodnoty rovný 2^0.

Cieľom použitia dvojkolových segmentov, kde jediné súčasné vysielanie v oboch kolách nastáva iba v prvom segmente podfázy, je zabrániť tomu, aby susedné vrcholy vrcholov v stave *neaktívny* (t.j. v podfáze j vrcholy z množiny $P_j^{-1} = S_{(j+2) \bmod 3}$ v stave *neaktívny*) prijatie vysielania v druhom alebo štvrtom segmente pôvodne smerovaného z P_j^{+1} do P_j^0 nechápali ako vysielanie v prvom segmente podfázy, ktoré je určené práve im. To by malo za následok chybnú synchronizáciu týchto vrcholov v rámci prebiehajúcej fázy.

Poznamenajme ďalej, že vrchol nemusí vždy získať správny odhad ohraničenia najväčšej priradenej hodnoty od svojich susedov v nasledujúcej vrstve. Nejaký zo susedov totiž mohol vstúpiť do stavu *vysielajúci* na popud nejakého vrcholu v inej fáze. Za ten čas už mohol vysielať kontaktnú správu v stave *vysielajúci* v niekoľkých fázach, či dokonca mohol stihnúť prejsť do stavu *neaktívny*. Vrchol v stave *prijímajúci*, tak získava skreslený odhad, ktorý je ale vždy menší alebo rovný, než maximálny skutočný odhad susedných vrcholov. Ukážeme, že toto skresľovanie nemá vplyv na to, že sa správny odhad dostane k zdrojovému vrcholu. Nech teda vrchol drží ako svoj odhad hodnotu, ktorá je správnym odhadom najväčšej priradenej hodnoty v sieti. Potom ale podľa algoritmu minimálne jeden jeho sused v predchádzajúcej vrstve, konkrétne ten, ktorý spôsobí jeho prechod do stavu *vysielajúci*, prijme tento odhad správne. Dôvodom je, že tento vrchol bude v stave *prijímajúci* počas celého vysielania správneho odhadu. Správny odhad ohraničenia najväčšieho identifikátora v sieti sa tak postupne dostane po vrstvách až k zdrojovému vrcholu. Formálne to možno ľahko ukázať indukciou po vrstvách. Zdrojový vrchol môže túto hodnotu následne odvysielať algoritmom *ENCODED-BROADCAST* ostatným vrcholom v sieti.

Celková časová zložitosť tohto algoritmu je $O(D\log Z)$, kde Z je odhad najväčšej priradenej hodnoty nejakému vrcholu v sieti. Najdlhšia „vetva“ má totiž dĺžku $O(D)$ a najdlhší vysielaný odhad je $O(\log Z)$.

Pozrime sa teraz, ako môžeme odstrániť predpoklad, že vrcholy poznajú svoje zaradenie do vrstvy. Tento predpoklad možno odstrániť tak, že algoritmus skombinujeme s algoritmom na určenie príslušnosti vrcholu k vrstve z predchádzajúcej vety. V ňom majú jednotlivé fázy presne stanovenú dĺžku trvania, čo ako sme konštatovali, umožňuje naviac aj relatívnu synchronizáciu aktuálneho čísla kola vzhľadom na začiatok algoritmu. Algoritmus na zaradenie vrcholov do vrstiev pracuje vo fázach známej dĺžky, pričom fáza k pracuje s vrcholmi vo vrstve L_k a L_{k+1} . Zatiaľ skonštruovaný algoritmus na určenie odhadu najväčšej priradenej hodnoty aktivuje vrcholy na vykonávanie algoritmu tiež po vrstvách, konkrétne v 1. segmente podfáz. Musíme však dávať pozor, aby tento algoritmus nepracoval s vrcholmi, ktoré ešte nemajú určenú príslušnosť k vrstve. Obe algoritmy môžeme skombinovať následovným spôsobom. Zdrojový vrchol najprv začne vykonávať algoritmus na zaradenie vrcholov do vrstiev upravený tak, že vrcholy ktoré ukončia vysielanie vo fáze k prejdú do stavu *neaktívny* algoritmu na určovanie odhadu. Zdrojový vrchol po ukončení fázy 0 algoritmu *ENCODED-LAYERING* začína vykonávať algoritmus na určovanie odhadu. Je treba dbať na to, aby obe algoritmy neinterferovali. Prvý segment každej podfázy preto upravíme tak, že vrchol v stave *oznamujúci* nevysiela kontaktnú správu a ani nemení svoj stav, ak je v takej vrstve a takom kole, že počas kôl prvého segmentu v nasledujúcich dvoch vrstvách ešte beží algoritmus *ENCODED-LAYERING*. To, či v nasledujúcej vrstve beží algoritmus *ENCODED-LAYERING* je možné vypočítať na základe čísla vrstvy a aktuálneho kola algoritmu. Aktuálne kolo algoritmu môže byť určené modifikovaným algoritmom *ENCODED-LAYERING*, prípadne môže byť vypočítané na základe vrstvy, do ktorej patrí vrchol vzhľadom na kolo, kedy prejde vrchol do stavu *oznamujúci*.

Celková časová zložitosť kombinovaného algoritmu je $O(D\log D + D\log Z)$ kôl. ♣

Tvrdenie 3.3.6: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý počas* $O(D \log D)$ *kôl vypočíta v zdroji siete také ohraničenie Z, že pre excentricitu zdroja D platí, že* $Z/2 < D \leq Z$.

Dôkaz (Algoritmus ENCODED-D-BOUND): Použije sa modifikácia algoritmu *ENCODED-FIND-BOUND*, kde hodnota priradená vrcholu siete bude jeho vzdialenosť od zdroja (číslo vrstvy). ♣

Tvrdenie 3.3.7: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý počas* $O(D \log n)$ *kôl určí v zdroji také ohraničenie Z, že pre najväčší použitý identifikátor vrcholu v sieti* ID_{MAX} *platí, že* $Z/2 < ID_{MAX} \leq Z$.

Dôkaz (Algoritmus ENCODED-ID-BOUND): Algoritmus je modifikáciou algoritmu *ENCODED-FIND-BOUND*, pričom ako hodnota priradená vrcholu sa použije jeho identifikátor, t.j. $Value(v) := ID(v)$. Keďže pre maximálny použitý identifikátor platí $ID_{MAX} = \max\{ID(v) \mid v \in V\} = O(n)$ a $O(D) \subseteq O(n)$, je celková zložitosť algoritmu $O(D \log n)$. ♣

I keď algoritmus uvedený v nasledujúcej vete nie je najefektívnejší, demonštruje niektoré možnosti použitia algoritmov z predchádzajúcich viet.

Tvrdenie 3.3.8: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý realizuje ARB počas* $O(n + D \log n)$ *kôl.*

Dôkaz (Algoritmus ENCODED-BOUND-BROADCAST): Algoritmus pozostáva z niekoľkých častí zodpovedajúcich algoritmom z predchádzajúcich viet.

1. Algoritmom *ENCODED-D-BOUND* (tvrdenie 3.3.6) zistí zdrojový vrchol ohraničenie parametra *D* siete počas $O(D \log D)$ kôl.
2. Algoritmom *ENCODED-ID-BOUND* (tvrdenie 3.3.7) zistí zdrojový vrchol ohraničenie najväčšieho použitého identifikátora v sieti počas $O(D \log n)$ kôl.
3. Algoritmom *ENCODED-BROADCAST* (tvrdenie 3.3.1) distribuuje zdrojový vrchol získané odhady parametrov ID_{MAX} a *D*. Vrcholy, po spoznaní ohraničenia

parametrov ID_{MAX} a D na základe kola, v ktorom tieto parametre prijali, vedia určiť kolo, kedy bude algoritmus *ENCODED-BROADCAST* trvajúci $O(D\log(n+D)) = O(D\log n)$ kôl ukončený, pretože posielaná správa je bitový zápis čísel ohraničení parametrov ID_{MAX} a D.

4. Vrcholy v počas $O(n)$ kôl – presnejšie počas Z kôl, kde Z je ohraničenie parametra ID_{MAX}, spoznajú identifikátory svojich susedov následovne: V kole i ($1 \le i \le Z$) tejto časti vysiela správu so svojim identifikátorom vrchol s identifikátorom i.
5. DFS prehľadávaním grafu s použitím tokenu, ktorý obsahuje zdrojovú správu, sa počas maximálne $2Z = O(n)$ kôl realizuje samotný broadcasting. Po prijatí tokenu musí vrchol správou oznámiť susedom, že už zdrojovú správu prijal, aby mu viac token nebol posielaný.

Ako vyplýva z času trvania jednotlivých častí algoritmu, celková zložitosť je $O(n + D\log n)$ kôl. ARB je realizované vďaka tomu, že po realizovaní prvých 3 častí algoritmu už všetky vrcholy poznajú čas ukončenia algoritmu dopredu. ♣

Tvrdenie 3.3.9: *Nech $G = (V, E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Potom existuje konštruktívny distribuovaný algoritmus, ktorý realizuje ARB počas $O(n + D\log D)$ kôl.*

Dôkaz (Algoritmus ENCODED-BOUND-BROADCAST-2): Algoritmus realizuje broadcasting rýchlejšie vďaka tomu, že neodhaduje znalosť parametru n a funguje následovne:

1. Algoritmom *ENCODED-D-BOUND* (tvrdenie 3.3.6) zistí zdrojový vrchol ohraničenie parametra D siete počas $O(D\log D)$ kôl.
2. Algoritmom *ENCODED-BROADCAST* (tvrdenie 3.3.7) distribuuje zdrojový vrchol získaný odhad parametra D. Vrcholy, po spoznaní ohraničenia parametra D a na základe kola, v ktorom tento parameter prijali, vedia presne určiť kolo, kedy najneskôr bude algoritmus *ENCODED-BROADCAST* trvajúci $O(D\log D)$ kôl ukončený. Poznamenajme, že táto časť môže byť optimalizovaná na čas trvania $O(D\log\log D)$ kôl, pretože vysielané ohraničenie Z parametra D je mocninou čísla 2, t.j. $Z = 2^z$ a teda vysielaná správa môže byť redukovaná na binárny zápis čísla z.
3. Algoritmom *EXPLORE-EXPAND-CHECK* z tvrdenia 3.3.3 realizujúcim distribuovaný ARB počas $O(n)$ kôl v symetrických sieťach s detekciou kolízie za predpokladu spontánnych vysielaní realizujeme broadcasting. Predpoklad spontánnych vysielaní je splnený, pretože po vykonaní 1. a 2. časti algoritmu sú všetky vrcholy pripravené začať v spoločnom kole vykonávať tento algoritmus.

Celková časová zložitosť vyplýva z časovej zložitosti jednotlivých častí algoritmu a je rovná $O(n + D\log D)$ kôl. ♣

Všimnime si, že aplikovaním prvých 2 časti predchádzajúceho algoritmu vieme ľubovoľný algoritmus pre distribuované symetrické rádiové siete s detekciou kolízie predpokladajúci spontánne vysielania a realizujúci komunikačnú úlohu počas $T(n)$ kôl prerobiť na algoritmus bez použitia spontánnych vysielaní, pričom časová zložitosť takto upraveného algoritmu bude $T(n) + O(D\log D)$ kôl.

Tvrdenie 3.3.10: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete s detekciou kolízie a informácia v každom uzle siete je obmedzená identifikátor vrcholu. Nech* A *je distribuovaný algoritmus riešiaci komunikačnú úlohu počas* $T(n)$ *kôl s využitím spontánnych vysielaní. Potom existuje algoritmus* A' *riešiaci komunikačnú úlohu bez využitia spontánnych vysielaní v čase* $T(n) + O(D\log D)$.

Kapitola 4

Broadcasting v 2D mriežke

Dvojrozmerná mriežka, ako topológia grafu dosiahnuteľnosti rádiovej siete, sa vyznačuje jednoduchosťou a veľkou mierou štruktúrovanosti. Často je preto predmetom štúdia pri navrhovaní rôznych distribuovaných algoritmov. V tejto kapitole sa budeme zaoberať otázkou návrhu efektívnych broadcastovacích algoritmov pri využití rôznych komunikačných scenárov v rádiových sieťach, ktorých topológiou je dvojrozmerná symetrická mriežka. Ukážeme, že vo všetkých uvažovaných komunikačných scenároch je možné vykonať broadcasting asymptoticky optimálne počas $O(D)$ kôl. Skonštruujeme algoritmus, ktorý vie broadcasting v uvedenom čase realizovať pre model s distribuovaným riadením, kde jediná informácia dostupná vrcholom je ich identifikátor. Pre jednotlivé komunikačné modely však skonštruujeme aj ďalšie broadcastovacie algoritmy, ktoré sú pre ne efektívnejšie než tento „všeobecný" algoritmus. Pod pojmom vrcholy okolo zdroja budeme v tejto kapitole chápať susedné vrcholy zdroja a vrcholy, ktoré majú ako susedné vrcholy dve susedné vrcholy zdroja. Symbolom Z budeme označovať hodnotu najväčšieho identifikátora použitého v sieti.

Definícia 4.0.1 (2D mriežka): *Mriežkou s rozmermi m x n nazývame neorientovaný graf* $G = (V, E)$, *kde*

$$V = Z_m \times Z_n = \{(i, j) \mid 0 \le i < m, 0 \le j < n\}$$
$$E = \{((i, j), (i', j')) \mid (i' = i \wedge j' = j \pm 1) \vee (j' = j \wedge i' = i \pm 1)\}$$

4.1 Pomocné procedúry

Definícia 4.1.1 (Vysielací reťazec vrcholu): *Nech* $a_k a_{k-1} \ldots a_0 = (ID(v))_2$ *je binárny zápis identifikátora vrcholu v rádiovej siete. Potom vysielacím reťazcom priradeným identifikátoru* $ID(v)$ *nazývame nekonečnú postupnosť* $Send(ID(v)) = (1, a_0, a_1, \ldots, a_{k-1}, a_k, 0, 0, \ldots, 0, \ldots)$.

Vysielací reťazec je nekonečná postupnosť, ktorá vznikne zreťazením čísla 1, obráteného binárneho zápisu identifikátora (t.j. zápis začína najnižším bitom identifikátora) a nekonečnej postupnosti núl. Vysielanie podľa vysielacieho reťazca umožňuje v istých špecifických podmienkach relatívne rýchly výber jedného zo susedných vrcholov.

Tvrdenie 4.1.1: *Nech* $ID_1, ID_2 \ge 1$ *(* $ID_1 \ne ID_2$ *) sú dva rôzne identifikátory vrcholov v rádiovej sieti, kde horným ohraničením všetkých identifikátorov je číslo Z. Potom vysielacie* *reťazce* $Send(ID_1)$ *a* $Send(ID_2)$ *sa líšia na pozícii i, kde* $i \leq \log Z + 2$*, t.j. na pozícii i má jeden z reťazcov hodnotu 1 a druhý hodnotu 0.*

Dôkaz: Dĺžka binárneho zápisu každého z identifikátorov je nanajvýš $\log Z + 1$. Uvažujme nenormalizované binárne zápisy čísel ID_1 a ID_2 zarovnané na dĺžku $\log Z + 1$. Keďže identifikátory ID_1 a ID_2 sú rôzne , ich binárne zápisy sa líšia aspoň na jednej pozícii. Z toho ako boli vysielacie reťazce konštruované už priamo vyplýva pravdivosť tvrdenia. ♣

Tvrdenie 4.1.2: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete. Nech* $v \in V$ *a* $u, w \in \Gamma_G(v)$ *sú jediní aktívni susedia vrcholu v. Potom v modely s distribuovaným riadením bez detekcie kolízie, kde jediná informácia vo vrchole je jeho identifikátor, vie vrchol v vybrať jedného z nanajvýš 2 aktívnych susedov počas* $O(\log n)$ *kôl.*

Dôkaz (Algoritmus BINARY-SEND): Pri návrhu algoritmu využijeme symetrickosť siete (grafu) a vysielací reťazec priradený identifikátoru spolu s jeho vlastnosťou uvedenou v tvrdení 4.1.1.

V 1. kole algoritmu vysiela vrchol *v* správu na začatie vysielania. V kole *2i* vysiela správu so svojim identifikátorom vrchol *u*, resp. *w* práve vtedy, ak na *i*-tej pozícii vo vysielacom reťazci priradenom jeho identifikátoru je hodnota 1. Ak vrchol *v* prijme v kole *2i* nejaký identifikátor (správu), v kole *2i+1* vysiela správu o ukončení algoritmu a zároveň už pozná identifikátor jedného zo svojich nanajvýš dvoch aktívnych susedov.

Keďže pre ohraničenie identifikátorov v sieti platí, že $Z = O(n)$, počas nanajvýš $O(\log n)$ kôl dôjde k tomu, že vysielacie reťazce sa budú líšiť a nastane úspešné (bezkolízne) odvysielanie správy s identifikátorom jedným z vrcholov *u* alebo *w*. ♣

Modifikáciou problému predchádzajúceho tvrdenia je problém, kedy dva vrcholy majú spoločný susedný vrchol, pričom chceme, aby aspoň jeden z týchto dvoch vrcholov doručil správu spoločnému susednému vrcholu. Opäť vysielanie podľa vysielacieho reťazca prislúchajúceho identifikátoru vrcholu vedie k úspešnému riešeniu úlohy. Všimnime si, že vysielací reťazec je navrhnutý tak, že k úspešnému doručeniu správy od jedného z vrcholov dôjde aj v prípade, keď susedné vrcholy začínajú podľa vysielacieho reťazca vysielať v rôznych kolách. Vtedy dochádza k úspešnému odvysielaniu hneď v prvom kole.

Je dôležité poznamenať, že v prípade, ak vysielanie podľa vysielacieho reťazca prislúchajúceho identifikátoru prevádzajú súčasne viac než 2 susedné vrcholy, môže mať za následok, že sa nepodarí úspešne odvysielať voči spoločnému (cieľovému) vrcholu žiadnu správu.

Tvrdenie 4.1.3: *Nech $G = (V, E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete. Nech $v \in V$. Potom v modely s distribuovaným riadením a detekciou kolízie, kde jediná informácia vo vrchole je jeho identifikátor, vieme vybrať jedného zo susedov vrcholu v počas $O(\log n)$ kôl.*

Dôkaz (Algoritmus Collision-Binary-Selection): Algoritmus je prepisom algoritmu *Binary-Selection* z tvrdenia 3.2.10 pre model s detekciou kolízie. Algoritmus pracuje vo fázach, z ktorých každá má 2 kolá. V každej fáze v 1. kole vrchol *v* vysiela interval $[a, b]$ a v 2. kole fázy vysielajú svoj identifikátor tie susedné vrcholy, ktorých identifikátor je v intervale $[a, b]$. Keďže sme v modely s detekciou kolízie, vrchol *v* vie určiť, či v 2. kole vysiela žiaden (šum pozadia), jeden (príjem správy) alebo viac ako jeden susedný vrchol (interferenčný šum).

V prvej časti algoritmus určí horné ohraničenie identifikátorov susedných vrcholov tak, že postupne vo fáze *i* vysiela vrchol *v* interval $[2^{i-1}, \infty]$ dovtedy, kým aspoň jeden sused má svoj identifikátor v tomto intervale. Ak vo fáze *j* žiaden sused nemá identifikátor v intervale $[2^{j-1}, \infty]$ znamená to, že horným ohraničením identifikátorov susedov vrcholu *v* je $2^{j-1} = t$. Ak *Z* horné ohraničenie identifikátorov všetkých vrcholov, tak nájdenie horného ohraničenia identifikátorov susedných vrcholov vieme realizovať v čase $O(\log Z)$.

V druhej časti algoritmu vyberieme jeden z identifikátorov susedov binárnym výberom (ak vrchol *v* má vôbec nejakých susedov). Nech $[a, b]$ je aktívny interval zdroja v danej fáze. V každej fáze zistíme, či má zdroj v intervale $[a, (a+b)/2]$ nejakého suseda. Ak áno, tak ako aktívny interval pre ďalšiu fázu si nastavíme $[a, (a+b)/2]$ a pokračujeme v ďalšej fáze. Ak nie, ako aktívny interval pre ďalšiu fázu si nastavíme $[(a+b)/2, b]$. Keďže vždy si vyberáme interval, kde má aspoň jeden sused vrcholu *v* svoj identifikátor, po $O(\log Z)$ fázach nájdeme určite nejaký identifikátor suseda vrcholu *v*. Na začiatku druhej časti algoritmu ako iniciálny aktívny interval použijeme $[0, t]$, kde *t* je v prvej časti algoritmu zistené horné ohraničenie identifikátorov susedov zdroja.

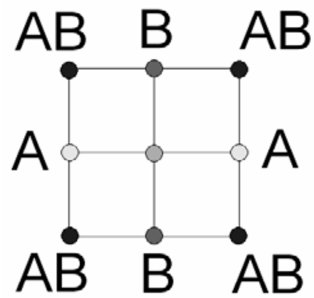
Prvá aj druhá časť algoritmu trvajú po $O(\log Z)$ dvojkolových fáz. Keďže $Z = O(n)$ je celková časová zložitosť algoritmu $O(\log n)$ kôl. ♣

4.2 Všeobecný broadcastovací algoritmus

Ukážeme najprv, že ak dokážeme určiť dve (resp. tri) špecifické triedy vrcholov okolo zdroja, ktorý môže byť kdekoľvek v mriežke, potom vieme vykonať broadcasting počas $O(D)$ kôl v modely bez detekcie kolízie a bez akejkoľvek lokálnej informácie vo vrchole (s výnimkou príslušnosti k jednej z tried v danom momente a niekoľko kôl dozadu). To znamená, že vieme vykonať broadcast optimálne na každom z uvažovaných modelov.

Algoritmus pracuje s množinou aktívnych vrcholov, ktoré v mriežke vytvárajú štvorec (ak by sme uvažovali nekonečnú mriežku), resp. časť štvorca. Ide vlastne o modifikáciu schémy, v ktorej vrchol hneď potom, čo prijme správu, posiela ju v nasledujúcom kole ako vysielač ďalej. Pri tejto schéme je jediný problém a to zabezpečiť doručenie správy do vrcholov, ktoré sa nachádzajú v rohoch vytváraného štvorca (vlny). Vhodným rozdelením vrcholov do tried vieme zabezpečiť, že ak vrchol bude vysielať v kole prislúchajúcom jeho triede, správa sa dostane i do rohových vrcholov. Zároveň pre ďalšiu vlnu aktívnych vrcholov vieme efektívne určiť ich zaradenie do niektorej z tried.

Predpokladajme teda, že máme k dispozícií nasledovné rozdelenie vrcholov okolo zdroja do tried:



Obrázok 4.2.1: Rozdelenie vrcholov do tried okolo zdroja

Ak sa zdroj nenachádza vo vnútri mriežky, ale napríklad niekde na okraji, budú prislúchajúce vrcholy v tejto inicializácii chýbať – to však nijako neovplyvní náš algoritmus.

Tvrdenie 4.2.1: *Nech $G = (V, E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete, ktorého topológiou je dvojrozmerná mriežka. Ak pre vrcholy okolo zdroja, ktoré poznajú zdrojovú správu, je známe rozdelenie do tried* ***A***, ***B*** *(podľa obrázku 4.2.1, v rovnakej triede sú susedné vrcholy zdroja, ktoré sú v mriežke oproti sebe) a* ***AB***, *potom existuje adaptívny konštruktívny algoritmus realizujúci rádiový broadcasting v sieti popísanej grafom G počas $O(D)$ kôl v modely s distribuovaným riadením, bez detekcie kolízie a bez akejkoľvek lokálnej informácie vo vrchole.*

Dôkaz (Algoritmus Grid-Broadcast): Algoritmus pracuje vo fázach, z ktorých každá sa skladá zo 6 kôl. Na začiatku algoritmu sú aktívne vrcholy triedy *A*, *B* a *AB* okolo zdroja, ktoré podľa predpokladu už poznajú zdrojovú správu σ. S každou správou vyslanou algoritmom sa vysiela aj číslo kola aktuálnej fázy, v rámci ktorého bola vyslaná. To zabezpečí, aby na správu v 5. a 6. kole fázy reagovali len tie vrcholy, pre ktoré je určená. Všetky vrcholy, ktoré v danom kole nevysielajú, vystupujú ako prijímače.

Vysielanie počas každej fáz je definované následovne:

1. kolo: Aktívne vrcholy triedy *A* vysielajú správu σ a typ triedy *A*

2. kolo: Aktívne vrcholy triedy *B* vysielajú správu σ a typ triedy *B*

3. kolo: Aktívne vrcholy triedy *AB* vysielajú správu σ a typ triedy *AB*

4. kolo: Vrcholy, ktoré prijali správu triedy *AB* v 3. kole, vystupujú ako prijímače. Vrcholy, ktoré v kole 1. alebo 2. prijali správu vystupujú ako vysielače a vysielajú svoj typ triedy. Vrcholy triedy *AB* si na základe prijatého typu triedy zmenia svoju triedu na *A* alebo *B*.

5. kolo: Vrcholy triedy *A*, ktoré v 4. kole menili svoj typ triedy (t.j. boli predtým triedy *AB*) vysielajú správu σ a typ triedy *A*.

6. kolo: Vrcholy triedy *B*, ktoré v 4. kole menili svoj typ triedy (t.j. boli predtým triedy *AB*) vysielajú správu σ a typ triedy *B*.

Vrcholy, ktoré prijmú správu v 5. a 6. kole si nastavia svoju triedu na *AB*. Aktívne vrcholy pre ďalšiu fázu sú všetky vrcholy, ktoré prijali správu v 1., 2. či 3. kole predchádzajúcej fázy alebo v 5. a 6. kole predchádzajúcej fázy prijali správu a patria do triedy *AB*. Vrcholy, ktoré prijali správu v 4. kole na ňu nereagujú. Všetky aktívne vrcholy v predchádzajúcej fáze sa stanú trvalo pasívnymi, t.j. na žiadne ďalšie správy už nereagujú.

Z uvedeného algoritmu je zrejmé, že v prípade, keď máme k dispozícii uvedené iniciálne rozdelenie vrcholov (ktoré už prijali zdrojovú správu) okolo zdroja do tried, vieme vykonať broadcasting v čase $O(D)$ bez ohľadu na rozmery mriežky, pozíciu zdroja v mriežke, ba dokonca aj bez znalosti identifikátorov jednotlivých vrcholov. ♣

4.3 Inicializačné algoritmy pre broadcastovací algoritmus

Teraz nám ostáva už len nájsť algoritmy, ktoré zabezpečia úvodné iniciálne rozdelenie vrcholov do tried a zároveň, ktoré zaručia, že všetkým vrcholom okolo zdroja bude správa doručená. Vo všetkých uvádzaných algoritmov budeme vždy so správou posielať aj číslo kola, v rámci ktorého bola odoslaná. Ako základ predpokladáme model s distribuovaným riadením, kde jediná informácia vo vrchole je jeho identifikátor. V tejto podkapitole predkladáme čitateľovi súbor možných inicializačných algoritmov pre algoritmus *Grid-Broadcast* z tvrdenia 4.2.1.

Algoritmus Grid-Broadcast-Init (1)

Ak zdroj pozná identifikátor aspoň jedného zo svojich susedov, vieme zaručiť inicializáciu algoritmu *Grid-Broadcast* v konštantnom čase (6 kôl) na každom z uvažovaných modelov následovne uvedeným algoritmom. Zopakujme, že súčasťou každej vyslanej správy je aj kolo, v rámci ktorého bola správa odoslaná. Všetky vrcholy, ktoré v algoritme nie sú v danom kole uvedené ako vysielajúce, vystupujú ako prijímače.

1. kolo: Zdroj odvysiela správu σ a identifikátor suseda, ktorého identifikátor pozná

2. kolo: Vrchol s odvysielaným identifikátorom si nastaví svoju triedu na *A* a vysiela správu σ. Vrcholy, ktoré túto správu prijali (okrem zdroja), si nastavia ako svoju triedu *AB*.
3. kolo: Vrcholy, ktoré sú v triede *AB* vysielajú správu σ. Vrcholy, ktoré prijali správu v 1. kole a zároveň aj teraz v 3. kole si nastavia svoju triedu na *B*.
4. kolo: Vrcholy triedy *B* vysielajú správu σ. Vrcholy, ktoré v 4. kole prijímajú správu σ po prvý krát, si nastavia svoju triedu na *AB*. Zároveň aj vrchol (je najviac jeden), ktorý v 1. kole prijímal správu a doteraz už nič neprijal si nastaví svoju triedu na *A*.
5. kolo: Vrcholy triedy *A* vysielajú správu σ.
6. kolo: Vrcholy triedy *B* vysielajú správu σ. Vrcholy, ktoré v 5. aj 6. kole prijímajú správu si nastavia svoju triedu na *AB* (okrem zdroja a vrcholov, ktoré v 5. a 6. kole vysielajú) a všetky ostatné vrcholy nastavia svoju triedu za neidentifikovanú a „zabudnú“ na to, že nejaké správy dostali. Vrcholy triedy *A*, *B* a *AB* sa stanú aktívnymi.

Ako je ľahko vidieť, tento algoritmus skutočne prevedie úvodnú inicializáciu potrebnú pre algoritmus *Grid-Broadcast*.

Z doterajšieho môžeme povedať, že v každom modely, kde zdroj pozná identifikátor aspoň jedného suseda, pričom detekciu kolízie nepredpokladáme, vieme realizovať broadcasting na mriežke v čase $\mathrm{O}(D)$. Ide teda napríklad o centralizovaný broadcasting, či broadcasting za predpokladu, že každý vrchol grafu pozná nejakú časť svojho grafového okolia.

Algoritmus Grid-Broadcast-Init (2)

V prípade modelu, kde každý vrchol pozná len svoj identifikátor a nemáme detekciu kolízie, môžeme ľahko postupovať následovne:

1. zdroj odvysiela správu σ v 1. kole
2. vrchol s identifikátorom i vysiela svoje ID v kole číslo $2i$, ak medzičasom neprišiel od zdroja signál ukončenia. Signál ukončenia vyšle zdroj v kole $2i+1$, ak v kole $2i$ prijal správu s identifikátorom i, t.j. zdroj pozná identifikátor niektorého zo susedov.

Ak predpokladáme, že identifikátory vrcholov sú čísla $1..Z$, kde $Z = O(n)$, tak v čase $O(n)$ bude zdroj poznať identifikátor jedného zo susedov. Pomocou *Grid-Broadcast-Init (1)* a *Grid-Broadcast* sa zabezpečí broadcasting v celkovom čase $O(n)$, pretože

$D = O(n)$. Neskôr ale ukážeme, že broadcasting je možné vykonať v tomto modely v čase $O(D + \log Z)$, kde Z je už spomínané horné ohraničenie identifikátorov vrcholov.

Algoritmus Grid-Broadcast-Init (3)

Ak máme k dispozícii detekciu kolízie môže zdroj previesť výber jedného so susedov v čase $O(\log n)$ pomocou algoritmu binárneho výberu *Collision-Binary-Selection* z tvrdenia 4.1.3. Celkovo pre tento model máme čas broadcastingu na mriežke $O(D + \log n)$ kôl s využitím predchádzajúcich algoritmov *Grid-Broadcast-Init(1)* a *Grid-Broadcast*.

Algoritmus Grid-Broadcast-Init (4)

Pre model bez detekcie kolízie, kde každý vrchol naviac okrem svojho identifikátora pozná aj horné ohraničenie identifikátorov *Z*, vieme vykonať inicializáciu v čase $O(\log^2 Z)$ kôl pomerne jednoduchým algoritmom, ktorý je modifikáciou myšlienky algoritmu *Binary-Send* z tvrdenia 4.1.2. Ten, ako sme ukázali, zaručí pri symetrickosti siete, že ak nanajvýš 2 vrcholy vysielajú správu pre ten istý vrchol, tak správa aspoň jedného z nich bude doručená v čase $O(\log Z)$ kôl. Uvedený algoritmus funguje i pri neznalosti ohraničenia *Z*. Znalosť *Z* nám umožňuje ukončiť vysielanie podľa vysielacieho reťazca i pri nesymetrickom modely, či inom modely, kde nemôžeme počítať s potvrdením prijatia správy od cieľového vrcholu.

Algoritmus je inicializovaný správou od zdroja. Po jej prijatí nasleduje najviac $2 + \lfloor \log Z \rfloor$ fáz. Vo fáze *i* sa správa vrchol (sused zdroja) podľa *i*-tej hodnoty vo vysielacom reťazci priradenom jeho identifikátoru – túto hodnotu budeme nazývať zvolenou hodnotou vrcholu pre danú fázu. Ku každej fáze môžeme priradiť binárny vektor podľa zvolených hodnôt susedov zdroja. Každá fáza sa skladá z dvojkolových segmentov. Prvé kolo segmentu je vysielacie kolo (nepárne kolá), druhé kolo je overovacie kolo (párne kolá). V overovacích kolách všetky vrcholy okrem zdroja vystupujú ako prijímače. Ak zdroj v predchádzajúcom vysielacom kole rozpoznal nejaký z identifikátorov susedného vrcholu, v overovacom kole vysiela signál o ukončení výberu, čím naštartuje algoritmus *Grid-Broadcast-Init (1).* Opisovať budeme preto len vysielacie kolá segmentov. V prvom segmente fázy vysielajú vo vysielacom kole svoj identifikátor tí susedia zdroja, ktorí majú pre danú fázu zvolenú hodnotu 1. Vo vysielacom kole 2. segmentu fázy vysielajú tí susedia, ktorí si pre danú fázu vybrali 0. Takto, ak vo vektore zvolených hodnôt susedov má práve jeden vrchol hodnotu 1 alebo 0, zdroj rozpozná jeho identifikátor. Ak sa tak nestalo (do konca 2. segmentu fázy zdroj neukončil výber), znamená to, že mohli nastať 3 možnosti:

1. všetci susedia zdroja (nie nutne 4 vrcholy) majú pre danú fázu zvolenú hodnotu 1
2. všetci susedia zdroja majú pre danú fázu zvolenú hodnotu 0

3. práve 2 susedné vrcholy majú pre danú fázu zvolenú hodnotu 1 a iné 2 vrcholy majú zvolenú hodnotu 0.

Po úvodných dvoch segmentoch nasleduje $2+\lfloor \log Z \rfloor$ segmentov, počas ktorých vo vysielacom kole susedia so zvolenou hodnotou 1 vysielajú v správe svoj identifikátor na základe vysielacieho reťazca priradeného identifikátoru. Ak majú vybranú hodnotu 1 práve dvaja susedia dôjde k prijatiu identifikátora niektorého zo susedov zdroja zdrojovým vrcholom a tým k ukončeniu výberu. V opačnom prípade úspešné vysielanie môže, ale nemusí nastať. Pokiaľ nenastane ukončenie výberu počas $2+\lfloor \log Z \rfloor$ segmentov, prechádzajú všetky susedné vrcholy zdroja do následovnej fázy (to znamená, že všetci susedia mali vybranú rovnakú hodnotu). Keďže identifikátory susedných vrcholov sú rôzne, existuje fáza, v ktorej si všetci susedia nevybrali rovnakú hodnotu. Počas nej dôjde k úspešnému výberu jedného zo susedov zdroja. Časová zložitosť tohto výberového algoritmu je $O(\log^2 Z)$. Znalosť Z je dôležitá kvôli tomu, že bez nej by sme nevedeli určiť, či to, že zdroj nepotvrdzuje prijatie hodnoty je dôsledok toho, že všetci susedia si vybrali rovnakú hodnotu, alebo ešte v danej fáze práve dvaja vysielajúci susedia nedokončili svoje vysielanie podľa vysielacieho reťazca. Nebola by teda možnosť, ako sa dostať do ďalšej fázy v prípade neúspechu.

Zložitosť broadcastingu je s použitím tohto inicializačného algoritmu $O(D+\log^2 n)$ kôl.

4.4 Hlavný broadcastovací algoritmus pre mriežku

V tejto záverečnej podkapitole ukážeme broadcastovací algoritmus pre model bez detekcie kolízie, kde jediná dostupná informácia vo vrcholoch sú ich identifikátory. Algoritmus prevedie broadcasting v čase $O(D+\log n)$. Všimnime si, že tento algoritmus je rýchlejší než broadcastovací algoritmus využívajúci inicializačný algoritmus *Grid-Broadcast-Init (4)* uvedený v predchádzajúcej podkapitole, pričom sa predpokladá menšia informácia vo vrcholoch siete. Predchádzajúce broadcastovacie algoritmy pracovali tak, že najprv inicializačným algoritmom našli vhodný iniciálny stav (vhodné rozdelenie vrcholov okolo zdroja do tried) pre algoritmus *Grid-Broadcast*, ktorý potom spustili. Algoritmus, ktorý v nasledujúcom tvrdení skonštruujeme, postupne skúša rôzne rozdelenia vrcholov okolo zdroja do tried. Nedokáže však vždy rozpoznať, či aktuálne rozdelenie je vhodným iniciálnym rozdelením pre algoritmus *Grid-Broadcast*. Preto je *Grid-Broadcast* spúšťaný aj vtedy, keď nie je isté, že bude úspešný. Ukážeme ale, že raz sa určite podarí spustiť úspešný broadcasting, ktorého dôsledkom bude tiež to, že algoritmus *Grid-Broadcast* sa už vrcholmi okolo zdroja nespustí a komunikácia v sieti ustane. Využívanie viacerých „paralelných" spustení algoritmu *Grid-Broadcast* má za následok zvýšenie celkového počtu vysielaní v sieti, čo môže byť pri uvažovaní iných charakteristík efektívnosti algoritmov, než je počet kôl potrebných na realizáciu úlohy, negatívny jav.

Tvrdenie 4.4.1: *Nech* $G = (V, E)$ *je neorientovaný graf dosiahnuteľnosti rádiovej siete, ktorého topológiou je dvojrozmerná mriežka. Potom existuje adaptívny konštruktívny algoritmus realizujúci rádiový broadcasting v sieti popísanej grafom G počas* $O(D + \log n)$ *kôl v modely s distribuovaným riadením a bez detekcie kolízie, kde jediná informácia vo vrchole je jeho identifikátor.*

Dôkaz (Algoritmus Main-Grid-Broadcast): Algoritmus využíva broadcastovací algoritmus *Grid-Broadcast* (tvrdenie 4.2.1) a čiastočne ideovo vychádza z algoritmu *Grid-Broadcast-Init (4)* (podkapitola 4.3).

Všimnime si, že algoritmus *Grid-Broadcast* pracuje vo fázach pozostávajúcich z konštantného a dopredu známeho počtu kôl (presnejšie každá fáza má 6 kôl). Vo fáze i ($i \geq 1$) vysielajú iba vrcholy vo vzdialenosti i od zdroja a niektoré vrcholy vo vzdialenosti $i+1$ od zdroja. Ak po uplynutí aspoň 3 fáz (18 kôl) algoritmu *Grid-Broadcast* spustíme susednými vrcholmi zdroja opäť tento algoritmus, máme zaručené, že jednotlivé inštancie algoritmu nebudú vzájomne interferovať. V algoritme musíme využívať upravenú verziu algoritmu *Grid-Broadcast* kvôli možnému vzniku rôznych „chybne“ inicializovaných inštancií tohto algoritmu. Zmena však spočíva len v pripojení informácie do každej poslanej správy o tom, v ktorej fáze algoritmu *Grid-Broadcast* bola správa vyslaná. Keď vrchol po prvý krát vysiela správu ako dôsledok vykonávania algoritmu *Grid-Broadcast* zapamätá si, v ktorej fáze bola táto správa vyslaná. Pri vykonávaní ďalších inštancií algoritmu *Grid-Broadcast* už potom nikdy nevysiela vo fázach vyšších než je táto fáza.

V algoritme *Main-Grid-Broadcast* rozlišujeme vrcholy dvoch typov: vrcholy okolo zdroja vykonávajúce výberový algoritmus a ostatné vrcholy siete vykonávajúce algoritmus *Grid-Broadcast*. Na začiatku algoritmu (v 0. kole) vysiela zdroj zdrojovú správu. Susedia zdroja po prijatí tejto správy začínajú vykonávať výberový algoritmus. Výberový algoritmus pracuje vo fázach pozostávajúcich z 3-kolových segmentov. Druhé kolo segmentu slúži na doručovanie správ o ukončení výberovej procedúry v dôsledku „naštartovania“ úspešného broadcastingu (algoritmu *Grid-Broadcast*). Sused zdroja, ktorý počas niektorého z kôl predchádzajúcej fázy prijme takúto správu, prestáva participovať na algoritme a ignoruje všetky prijaté správy. Tretie kolo segmentu je vyhradené pre správy, ktoré vysiela zdroj v prípade, že sa mu podarí v 1. kole segmentu v prijatej správe rozpoznať identifikátor jedného zo susedných vrcholov. Správa v 3. kole segmentu tak informuje susedov zdroja o ukončení výberovej procedúry a začiatku vykonávania algoritmu *Grid-Broadcast-Init (1)*. Rovnako ako v algoritme *Grid-Broadcast-Init (4)* bude vysielanie susedov zdroja v *i*-tej fáze závisieť od *i*-tej hodnoty vo vysielacom reťazci priradenom ich identifikátoru. Túto hodnotu budeme nazývať zvolenou hodnotou vrcholu pre danú fázu. Po skončení každej fázy sa bude čakať 3 fázy algoritmu *Grid-Broadcast* predtým, než sa vstúpi do ďalšej fázy výberového algoritmu.

V 1. kole 1. segmentu fázy vysielajú zdrojovú správu s identifikátorom susedia zdroja, ktorých zvolená hodnota pre danú fázu je 1. Podobne v 1. kole 2. segmentu fázy vysielajú zdrojovú správu s identifikátorom susedia zdroja, ktorých zvolená hodnota pre danú fázu je 0. Ak do ukončenia 2. segmentu fázy nebol výberový algoritmus ukončený, znamená to, že aspoň dva vrcholy siete majú pre danú fázu zvolenú rovnakú hodnotu. Ak má zdroj 4 susedov, tak práve dvaja z nich majú zvolenú hodnotu 1 a ďalší dvaja zvolenú hodnotu 0. Ak zdroj nemá 4 susedov, tak môžeme tvrdiť, že všetci susedia zdroja majú v danej fáze zvolenú rovnakú hodnotu. Zaoberajme sa prípadom, keď predpokladáme, že zdroj má 4 susedov. Susedné vrcholy zdroja s rovnakou zvolenou hodnotou môžu byť v mriežke umiestnené oproti sebe alebo pri sebe (vrcholy majú spoločný susedný vrchol rôzny od zdroja). Umiestnenie oproti sebe zodpovedá vhodnému rozdeleniu vrcholov do tried, no nevieme ho zistiť. Na druhej strane, umiestnenie pri sebe vieme počas 4 kôl (v zmysle výberového algoritmu počas 4 segmentov) zistiť následovne:

1. kolo: Vrcholy so zvolenou hodnotou 1 vysielajú správu *CHECK*
2. kolo: Vrcholy, ktoré v 1. kole prijali správu *CHECK* vystupujú ako vysielače a vysielajú správu prijatú v 1. kole
3. kolo: Vrcholy, ktoré v 2. kole prijali správu *CHECK* a sú zároveň susedmi zdroja so zvolenou hodnotou 0, vystupujú ako vysielače a vysielajú správu *CHECK-OK*
4. kolo: Vrcholy, ktoré v 3. kole prijímajú správu, vystupujú ako vysielače a vysielajú správu prijatú v 3. kole.

Po skončení vyššie opísaných 4 kôl susedia zdroja so zvolenou hodnotou 1, ktorí prijali správu *CHECK-OK* v 4. kole a susedia zdroja so zvolenou hodnotou 0, ktorí v 2. kole prijali správu *CHECK* vedia, že nastala situácia, kedy sú susedia zdroja s rovnakou hodnotou pri sebe. Susedia so zvolenou hodnotou 1 začnú postupne vysielať svoj identifikátor v 1. kole segmentov podľa vysielacieho reťazca priradeného ich identifikátoru. Po najviac $O(\log Z)$ segmentoch sa zdroj dozvie identifikátor jedného zo svojich susedov, čo vedie k ukončeniu výberového algoritmu (viď algoritmus *Binary-Send* z tvrdenia 4.1.2).

Ak po skončení prvých 6. segmentov fázy (2. segmenty vysielania, 4 segmenty overovania, či sú dvaja susedia so zvolenou hodnotou 1 pri sebe) nenastalo ukončenie výberovej procedúry, resp. jej ukončenie nie je isté (ako dôsledok toho, že dvaja susedia s hodnotou 1 sú pri sebe), znamená to, že mohli nastať 2 situácie:

- zdroj má 4 susedov, z ktorých majú dvaja zvolenú hodnotu 1 a dvaja zvolenú hodnotu 0, pričom susedia s rovnakou hodnotou sú oproti sebe
- všetci susedia zdroja majú zvolenú rovnakú hodnotu

Vzniknuté situácie nevieme z pohľadu susedov zdroja nijako rozlíšiť. Preto naštartujeme broadcasting tak, že susedia so zvolenou hodnotou 0 si ako svoju triedu určia A a susedia so zvolenou hodnotou 1 si ako svoju triedu určia *B*. V 7. segmente fázy vysielajú správu s inicializáciou broadcastingu vrcholmi triedy *A* susedia zdroja v triede *A* a podobne v 8. segmente susedia zdroja v triede *B*. Vrcholy, ktoré v 1. kole 7. aj 8. segmentu prijali správu si nastavia svoju triedu na *AB*. Ostatné vrcholy na prijatie správ zabudnú. Po 8. segmente fázy vrcholy okolo zdroja (vrcholy triedy *A*, *B* a *AB*) realizujú 1. fázu algoritmu *Grid-Broadcast*. Algoritmus *Grid-Broadcast* je prirodzene treba upraviť tak, aby jedno kolo zodpovedalo jednému segmentu – t.j. trvalo 3 kolá. Všimnime si, že ak všetci susedia zdroja majú zvolenú rovnakú hodnotu, tak počas 7. a 8. segmentu nevznikne žiaden vrchol triedy *AB*. Susedia zdroja však o vzniku (nevzniku) vrcholov triedy *AB* nevedia a preto vykonávajú algoritmus *Grid-Broadcast* bez ohľadu na tento fakt. Potenciálne vzniknuté chybné vykonávanie algoritmu *Grid-Broadcast* má ale tú vlastnosť, že realizuje broadcasting úplne rovnako ako vhodne inicializovaný *Grid-Broadcast*, až na to, že vrcholy ležiace na „uhlopriečke“ mriežky s priesečníkom v zdroji sa zdrojovú správu nedozvedia. Dokonca aj vrcholy vykonávajú algoritmus *Grid-Broadcast* vždy v rovnakých fázach. To znamená, že táto chybná inštancia algoritmu neovplyvní iné (predchádzajúce, či budúce) inštancie algoritmu. Naopak vznik aspoň jedného vrcholu triedy *AB* znamená, že aktuálne rozdelenie do tried je vhodné pre *Grid-Broadcast* a naštartovaná inštancia realizuje broadcasting. Preto vrcholy triedy *AB* okolo zdroja, ktoré pri takejto vhodnej inicializácii vznikli, začnú vysielať podľa vysielacieho reťazca svojho identifikátora správu o ukončení výberového algoritmu ako dôsledku naštartovania úspešného broadcastingu v 2. kole každého segmentu. Ak sa medzičasom v niektorej z ďalších fáz podarí naštartovať inú úspešnú inštanciu *Grid-Broadcast*, tak tieto vrcholy triedy *AB* na túto zmenu nereagujú (t.j. ak neodvysielali ešte všetky správy podľa vysielacieho reťazca, tak pokračujú ďalej vo vysielaní). Po $O(\log Z) = O(\log n)$ segmentoch sa každý zo susedov zdroja dozvie o ukončení výberovej procedúry. Každý sused zdroja má totiž dvoch susedov triedy *AB* a teda vysielanie podľa vysielacieho reťazca doručí správu o ukončení. Poznamenajme tiež, že susedia zdroja sa nemusia dozvedieť správu o ukončení výberovej procedúry v rovnakom čase, resp. v rovnakej fáze. Kým všetci susedia zdroja budú oboznámení o ukončení výberového algoritmu, môže včasná neaktivita niektorých z nich viesť k ďalším chybným inštanciám algoritmu *Grid-Broadcast*. Vďaka úprave algoritmu *Grid-Broadcast* tieto chybné inštancie však ale nikdy netrvajú dlhšie než korektné inštancie - teda $O(D)$ kôl. Totiž fáza prvej inštancie algoritmu *Grid*-Broadcast, ktorú nejaký vrchol vykonáva, je rovnaká ako fáza, v ktorej vykonáva inštanciu realizujúcu úspešný broadcasting. Zároveň žiaden vrchol v sieti nevykoná žiadnu inštanciu algoritmu *Grid-Broadcast* vo fáze vyššej, než je posledná fáza inštancie realizujúcej úspešný broadcasting.

Ako je zrejmé z činnosti algoritmu, uvedeným algoritmom vieme vykonať broadcasting na mriežke v celkovom čase $O(D + \log n)$ kôl. Poznamenajme, že zdroj sa o realizovaní úspešného broadcastingu nemusí vždy dozvedieť. ♣

Ako už bolo konštatované v úvode podkapitoly, algoritmus *Main-Grid-Broadcast* využíva viacero paralelne bežiacich inštancií algoritmu *Grid-Broadcast*, čo vedie k zvýšeniu počtu posielaných správ. To môže byť v niektorých aplikáciách nežiadúci jav. Poznamenajme preto, že je možné skonštruovať aj algoritmus, ktorý počas $O(\log n)$ kôl zabezpečí, že zdroj spozná identifikátor jedného zo susedov zdroja aj v modely, kde jediná dostupná informácia vo vrcholoch je identifikátor. Potom už algoritmom *Grid-Broadcast-Init (1)* je možná realizácia broadcastingu počas $O(D)$ kôl. Tento inicializačný algoritmus vychádza z výberovej procedúry algoritmu *Main-Grid-Broadcast*. Keďže jeho opis by bol technický zložitý, stručne opíšeme jeho hlavnú myšlienku. Všimnime si, že spúšťanie inštancií algoritmu *Grid-Broadcast* vždy, keď nastáva nerozhodná situácia, nie je nevyhnutné. V týchto nerozhodných situáciách úplne stačí skúsiť vytvoriť vrcholy triedy *AB* okolo zdroja. Vieme už, že ak je takéto vytvorenie vrcholov triedy *AB* úspešné, znamená to, že aktuálne rozdelenie vrcholov do tried je vhodné pre algoritmus *Grid-Broadcast.* O tom ale susedia zdroja nevedia. Ak je číslo prebiehajúcej fázy výberového algoritmu súčasťou správ, potom vrcholy triedy *AB* si môžu zapamätať číslo fázy, v ktorej boli vytvorené. Mechanizmom, aký je v algoritme *Main-Grid-Broadcast* použitý na oznámenie spustenia inštancie algoritmu *Grid-Broadcast* realizujúcej úspešný broadcasting vrcholmi triedy *AB* okolo zdroja, je možné následne susedom zdroja oznámiť, v ktorej fáze mali susedia zdroja zvolené také hodnoty, že určite práve 2 vrcholy mali zvolenú hodnotu 1. Susedné vrcholy zdroja si podľa čísla fázy a vysielacieho reťazca vedia určiť spomínanú zvolenú hodnotu pre danú fázu. Potom už len v na to vyčlenenom kole fázy susedia zdroja s vtedy zvolenou hodnotou 1 vysielajú podľa vysielacieho reťazca svoj identifikátor zdroju. Keďže tak robia nanajvýš dvaja susedia, vedie to k tomu, že zdroj prijme v správe identifikátor jedného zo svojich susedov. Opísaný inicializačný algoritmus tak spustí algoritmus *Grid-Broadcast-Init (1)* iba ak povedie k úspešnej realizácii broadcastingu.

Pre symetrickú mriežku a ľubovoľne zvolený zdrojový vrchol platí, že $O(\sqrt{n}) \subseteq O(D)$. Z toho dostávame, že $O(\log n) \subseteq O(\log^2 n) \subseteq O(D)$. Teda pre každý z modelov existuje broadcastovací algoritmus na mriežke pracujúci v čase $O(D)$, čo je asymptoticky optimálne vzhľadom na triviálny dolný odhad broadcastingu $\Omega(D)$.

Tvrdenie 4.4.2: *Nech $G = (V, E)$ je neorientovaný graf dosiahnuteľnosti rádiovej siete, ktorého topológiou je dvojrozmerná mriežka. Potom existuje adaptívny konštruktívny algoritmus realizujúci asymptoticky optimálne rádiový broadcasting v sieti popísanej grafom G počas $O(D)$ kôl v modely s distribuovaným riadením a bez detekcie kolízie, kde jediná informácia vo vrchole je jeho identifikátor.*

Kapitola 5

Broadcasting v planárnych grafoch

Táto kapitola je venovaná otázke rádiového broadcastingu v sieťach, ktorých graf dosiahnuteľnosti je planárny.

Definícia 5.1 (Planárny graf): *Orientovaný graf $G=(V,E)$ je planárny, ak sa dá nakresliť v rovine bez prekríženia hrán, t.j. jediné spoločné body hrán sú ich spoločné vrcholy.*

Oblasť rádiového broadcastingu v sieťach popísaných planárnym grafom dosiahnuteľnosti nebola nijako špeciálne skúmaná. Zrejme jedným z dôvodov je fakt, že mnohé dolné odhady času realizácie distribuovaného broadcastingu ako kontrapríkladové grafy využívajú práve planárne grafy (tvrdenia 3.2.5, 3.2.7). Pokiaľ je nám známe, na realizáciu distribuovaného rádiového broadcastingu v sieťach s planárnym grafom dosiahnuteľnosti zatiaľ neexistujú žiadne špecializované algoritmy. V prípade centralizovaného broadcastingu pre planárne grafy (presnejšie pre širšiu množinu grafov, ktorej podmnožinou sú planárne grafy) existuje algoritmus, ktorý generuje prijateľný rozvrh rádiového broadcastingu kratšej dĺžky než algoritmy pre všeobecné grafy. V práci [EKc04] Elkin a Kortsarz prezentovali algoritmus (tvrdenie 3.1.4), ktorý pre všeobecné neorientované grafy generuje prijateľné rozvrhy rádiového broadcastingu dĺžky $O(D+\log^4 n)$. Zároveň prezentovali aj modifikáciu tohto algoritmu, ktorá pre siete s grafom dosiahnuteľnosti nízkeho rádu, generuje rozvrhy menšej dĺžky. Konkrétne pre neorientované planárne grafy dosiahnuteľnosti platí nasledujúce tvrdenie:

Tvrdenie 5.1 [EKc04]: *Nech $G=(V,E)$ je neorientovaný planárny graf dosiahnuteľnosti rádiovej siete. Potom existuje deterministický algoritmus, ktorý v polynomiálnom čase pre daný zdrojový vrchol s vygeneruje prijateľný rozvrh Π realizujúci rádiový broadcasting taký, že $|\Pi| \in O(D+\log^3 n)$.*

V tejto kapitole prezentujeme deterministický algoritmus, ktorý pre orientované graf dosiahnuteľnosti $G=(V,E)$ a pre daný zdrojový vrchol $s \in V$ vygeneruje prijateľný rozvrh dĺžky $3.D$.

Pri tomto algoritme sa nebudeme vychádzať zo štruktúry planárneho grafu, ale z existencie planárneho nakreslenia grafu - t.j. pri konštrukcii sa budeme opierať o geometrickú realizáciu konkrétneho nakreslenia grafu v rovine. Ďalším dôležitým faktom je skutočnosť, že ak $G = (V, E)$ je planárny graf, potom každý jeho podgraf je planárny. Okrem uvedených faktov využijeme algoritmus, ktorý prezentoval Walter Schnyder v práci [S90].

Tvrdenie 5.2 [S90]**:** *Nech $G = (V, E)$ je planárny graf s $n \geq 3$ vrcholmi. Potom existuje jeho vloženie na mriežku $(n-2)\times(n-2)$, pričom jednotlivé hrany sú úsečky. Naviac toto vloženie je možné vypočítať v lineárnom čase.*

Ak si teda zoberieme ľubovoľný planárny graf $G = (V, E)$, potom vieme v lineárnom čase nájsť také priradenie, ktoré každému vrcholu grafu priradí bod v R^2 tak, že každej hrane grafu bude zodpovedať úsečka medzi bodmi priradenými vrcholom tejto hrany. Zároveň v tomto nakreslení nebudú existovať prekrížené hrany. Keďže uvažujeme centralizovaný broadcasting, zoberme a fixujme si nejaké takéto nakreslenie (napríklad vygenerované algoritmom z [S90]).

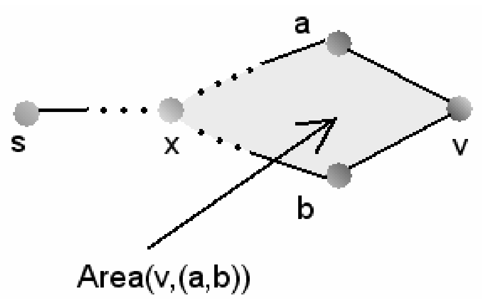
Predpokladajme pre jednoduchosť, že uvažovaný graf je symetrický (resp. neorientovaný). Ako však ukážeme neskôr, pre algoritmus nie je tento predpoklad potrebný.

Nech $v \in V$ je vrchol grafu $G = (V, E)$ a $s \in V$ je zdrojový vrchol. Označme si $l(v)$ dĺžku najkratšej (orientovanej) cesty z vrcholu s do vrcholu v. Na základe hodnoty $l(v)$ môžeme rozdeliť vrcholy grafu do $D+1$ disjunktných skupín (vrstiev) L_i, kde D je excentricita zdrojového vrcholu, t.j. $L_i = \{v \in V \mid l(v) = i\} = \{v \in V \mid d_G(s, v) = i\}$. Zrejme $L_o = \{s\}$ a $L_1 = \{v \in V \mid (s, v) \in E\}$. Ďalej nech P je funkcia, ktorá každému vrcholu $v \in L_i$ okrem zdroja priradí vrchol $P(v) \in L_{i-1}$ taký, že $(P(v), v) \in E$. Táto funkcia pre každý vrchol „určuje" jedinú neorientovanú cestu k zdroju: $SourcePath(v) = (v, P(v), P(P(v)), ..., s)$. Túto cestu pomenujeme ako *zdrojová cesta* vrcholu v. Poznamenajme, že toto priradenie môže byť ľubovoľné a nijako nesúvisí s posielaním správ v broadcastovacej schéme. Broadcastovacou schémou budeme v tejto kapitole nazývať konkrétnu realizáciu vysielania správ podľa vygenerovaného rozvrhu. Ak uvažujeme orientovaný graf, tak zdrojová cesta nemusí byť aj cestou v grafe. Na priradenie sa môžeme pozerať tiež ako na konštrukciu kostry grafu s koreňom v zdrojovom vrchole. Je zrejmé, že ľubovoľné dve zdrojové cesty majú spoločný aspoň jeden vrchol a to zdrojový vrchol. Ďalej, ak majú dve zdrojové cesty spoločné nejaký vrchol, tak cesta od tohto spoločného vrcholu k zdroju je pre obe zdrojové cesty rovnaká. Ďalšie pojmy zavedieme priamo pri opise konštrukcie.

Konštruovaná schéma bude posielať zdrojovú správu po vrstvách, pričom na doručenie správy z vrstvy L_i do každého vrcholu vrstvy L_{i+1} bude treba nanajvýš 4 kolá (neskôr ukážeme, že v skutočnosti vystačia 3 kolá). Na doručenie zdrojovej správy všetkým vrcholom vo vrstve L_1 stačí jedno vysielanie zdrojového vrcholu s. Zaoberať sa teda budeme vysielaním z vrstvy L_i do vrstvy L_{i+1} pre $i > 1$ za predpokladu, že všetky vrcholy vrstvy L_i už prijali zdrojovú správu.

Pozrime sa najprv na vrcholy vo vrstve L_{i+1} . Nech $v \in L_{i+1}$. Označme $S(v) = \{w \in L_i \mid (w,v) \in E\}$ - t.j. $S(v)$ je množina tých vrcholov vo vrstve L_i, ktoré môžu odvysielať správu vrcholu $v \in L_{i+1}$. Je jasné, že $|S(v)| > 0$. V prvom kole vysielania necháme každý vrchol vo vrstve L_i odvysielať zdrojovú správu. V tomto kole sa správu dozvedia tie vrcholy vo vrstve L_{i+1}, pre ktoré platí $|S(v)| = 1$. Týmito vrcholmi vrstvy L_{i+1} sa v ďalšom uvažovaní nemusíme zaoberať. Od tohto miesta budeme pod vrstvou L_{i+1} rozumieť iba tie vrcholy, ktoré neprijali správu v prvom kole, t.j. vrcholy s $|S(v)| \geq 2$. Ukážeme ako naplánovať vysielanie vrcholov vrstvy L_i v nasledujúcich 3 kolách tak, že každý vrchol vo vrstve L_{i+1} prijme správu aspoň raz.

Z predchádzajúceho máme, že sa zaoberáme len s vrcholmi , pre ktoré platí $|S(v)| \geq 2$. Zoberme si ľubovoľný vrchol $v \in L_{i+1}$ a dve rôzne vrcholy $a, b \in S(v) \subseteq L_i$. Nech x je prvý spoločný vrchol zdrojovej cesty vrcholu a a zdrojovej cesty vrcholu b - t.j. x je taký vrchol, ktorý sa nachádza na oboch zdrojových cestách tak, že cesta od tohto vrcholu x je v oboch zdrojových cestách rovnaká a spomedzi spoločných ciest je táto najdlhšia. Taký prvok x vždy existuje a je to prinajmenšom zdrojový vrchol. Potom cesta $(v, a, P(a), P(P(a)),..., x,...., P(P(B)), b, v)$ je cyklus a teda delí rovinu na 2 oblasti. Označme vnútornú oblasť symbolom $Area(v,(a,b))$. Pri fixovanom nakreslení a vzhľadom na jednoznačné určenie zdrojových ciest je táto oblasť pre danú trojicu vrcholov jednoznačne určená. Z geometrického hľadiska je tento útvar nejaký n-uholník s $n \geq 4$.



Obrázok 5.1: Určenie oblasti Area(v,(a,b)).

Ľahko možno ukázať niekoľko tvrdení o vzniknutých oblastiach:

$$\{a,b\} \subset Area(v,(a,b))$$
$$c,d \in S(v) \wedge c,d \in Area(v,(a,b)) \Rightarrow Area(v,(c,d)) \subseteq Area(v,(a,b))$$
$$c \in S(v) \wedge c \notin Area(v,(a,b)) \Rightarrow Area(v,(a,b)) \subset Area(v,(a,c)) \vee Area(v,(a,b)) \subset Area(v,(b,c))$$

Z posledného z uvedených tvrdení vyplýva, že ak máme nejakú oblasť A vrcholu v, v ktorej neleží nejaký vrchol $c \in S(v)$, tak vieme skonštruovať oblasť, v ktorej tento prvok leží a zároveň v nej ležia aj všetky prvky oblasti A. Teda postupným pridávaním prvkov z $S(v)$ do nejakej základnej oblasti vrcholu v, vieme skonštruovať takú oblasť, v ktorej ležia všetky prvky z $S(v)$. Presnejšie:

$$\exists\, a,b \in S(v) \textit{ také, že } S(v) \subset Area(v,(a,b)) = Area(v)$$

Pomenujme oblasť $Area(v)$ ako oblasť vrcholu v, prvky a,b ako hlavné susedné vrcholy vrcholu v a prvky z $S(v)-\{a,b\}$ ako vedľajšie susedné vrcholy vrcholu v.

Poznamenajme, že všetky doteraz uvádzané činnosti a určenie význačných vrcholov je možné vykonať algoritmicky v polynomiálnom čase.

Pozrime sa teraz na oblasti prislúchajúce rôznym vrcholom vrstvy L_{i+1}.

Lema 5.1: *Nech* $v,w \in L_{i+1}$, $v \neq w$ *a* $v \in Area(w)$, *potom platí* $Area(v) \subset Area(w)$.

Dôkaz: Nech $v,w \in L_{i+1}$, $v \neq w$ a $v \in Area(w)$. Potom pre každé $a \in S(v)$ cesta $(v, SourcePath(a))$ leží celá v $Area(w)$. Ak by totiž nemala celá ležať v $Area(w)$, musel by aspoň jeden jej bod ležať mimo oblasti $Area(w)$. Keďže graf je planárny, mimo sa cesta môže dostať len tak, že prejde cez vrchol $u \in V$ ležiaci na hranici oblasti $Area(w)$. Každý vrchol na hranici oblasti $Area(w)$ okrem w leží na nejakej zdrojovej ceste, pretože samotná hranica tejto oblasti je tvorená nejakými dvoma zdrojovými cestami. Aj $SourcePath(a)$ je zdrojová cesta a teda od vrcholu u musí byť totožná so zdrojovou cestou tvoriacou hranicu oblasti $Area(w)$, t.j. $SourcePath(a)$ musí ležať vo vnútri oblasti alebo na hranici oblasti $Area(w)$. Každopádne však celá leží v oblasti $Area(w)$, lebo aj hranica oblasti je jej časťou. Ostal nám, ešte prípad, že $SourcePath(a)$ prechádza cez hranicu oblasti $Area(w)$ v bode w. To ale nie je možné, lebo cez vrchol $w \in L_{i+1}$ žiadna zdrojová cesta vrcholu z L_i viesť nemôže (vzhľadom na definíciu zdrojovej cesty). Poznamenajme, že ak by sme chceli byť úplne formálne správny, tak správna definícia oblasti by mala okrem oblasti „n-uholníka" zahrňovať aj cestu od spojenia dvoch zdrojových ciest, ktoré ju vytvárajú, do zdrojového vrcholu. Keďže oblasť $Area(v)$ je ohraničená nejakými dvoma zdrojovými cestami, ktoré sú z vyššie uvedeného určite v $Area(w)$ a $v \in Area(w)$, musí byť aj celá oblasť $Area(v)$ v $Area(w)$, t.j. $Area(v) \subseteq Area(w)$. Naviac platí, že $Area(v) \neq Area(w)$. Ak by boli totiž oblasti rovnaké, museli by mať rovnaké hranice, ktoré sú tvorené (neorientovanými) hranami grafu. Vrchol v leží z definície oblasti na hranici oblasti $Area(v)$, a analogicky aj vrchol w na hranici oblasti $Area(w)$. Zhoda oblastí znamená aj zhodu hraníc. Pri zhode hraníc by platilo, že v je na hranici oblasti $Area(w) = Area(v)$. Hranica $Area(w)$ je tvorená z definície vrcholom w ($w \neq v$) a vrcholmi z vrstiev L_j pre $j \leq i$. Vrchol v však je z vrstvy L_{i+1}, čo je spor s tým, že leží na hranici oblasti $Area(w)$. Teda $Area(v) \subset Area(w)$. ♣

Lema 5.2: *Existuje vrchol* $v \in L_{i+1}$ *taký, že neexistuje vrchol* $w \in L_{i+1}$ *s vlastnosťou* $v \in Area(w)$.

Dôkaz: Je zrejmé, že ak $|L_{i+1}| \leq 2$, tak existenciu prvku máme z vyššie uvedeného zaručenú. Nech teda $|L_{i+1}| \geq 3$. Predpokladajme sporom, že prvok $v \in L_{i+1}$ s danou vlastnosťou neexistuje. To znamená, že pre každý prvok $u \in L_{i+1}$ existuje prvok $u' \in L_{i+1}$ taký, že $u \neq u' \wedge u \in Area(u')$. Vezmime si ľubovoľný prvok $v_1 \in L_{i+1}$. K nemu existuje $v_2 \in L_{i+1}$ s vlastnosťou $v_1 \neq v_2 \wedge v_1 \in Area(v_2)$. Z predošlej lemy máme, že $Area(v_1) \subset Area(v_2)$. Predpokladajme, že sme už určili vrcholy $v_1, ..., v_k$. Pre vrchol $v_k \in L_{i+1}$ na základe sporného predpokladu existuje prvok $v_{k+1} \in L_{i+1}$ taký, že $v_k \neq v_{k+1} \wedge v_k \in Area(v_{k+1})$, z čoho vyplýva $Area(v_k) \subset Area(v_{k+1})$. Ak by platili, že $v_{k+1} = v_j$ pre nejaké $j \in \{1,...,k\}$, dostali by sme, že $Area(v_j) \subset Area(v_{j+1}) \subset ... \subset Area(v_k) \subset Area(v_{k+1}) = Area(v_j)$. Z toho vyplýva $Area(v_j) \neq Area(v_j)$, čo je spor. Ak ale uvážime tento fakt a definovanú spornú vlastnosť, tak máme, že pre každé $v_k \in L_{i+1}$ vieme nájsť $v_{k+1} \in L_{i+1}$ s vlastnosťou $v_k \neq v_{k+1} \wedge v_k \in Area(v_{k+1})$, ktoré je naviac rôzne od všetkých predchádzajúcich určených vrcholov. Takto, ale môžeme skonštruovať nekonečnú postupnosť vrcholov z L_{i+1}, čo je spor, pretože množina L_{i+1} je konečná. ♣

Definujme teraz množiny:

$$M_0 = \{v \in L_{i+1} \mid \nexists\, w \in L_{i+1} \text{ taký, že } v \in Area(w)\}.$$

$$N_k = \{v \in L_{i+1} \mid v \notin \bigcup_{j=0}^{k-1} M_j\}$$

$$M_k = \{v \in N_k \mid \nexists\, w \in N_k \text{ taký, že } v \in Area(w)\}$$

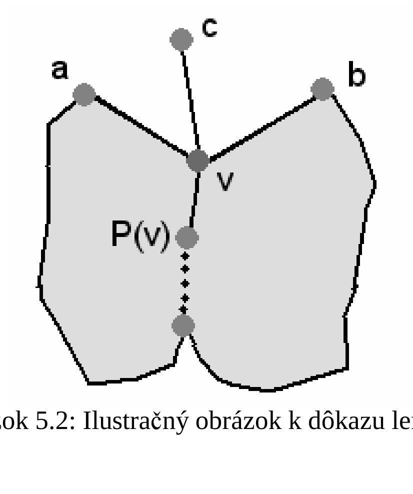
Množina M_0 sú tie vrcholy vrstvy L_{i+1}, ktoré sa nenachádzajú v oblasti žiadneho iného vrcholu vrstvy L_{i+1}. Z geometrického pohľadu tvoria spolu s hranami do hlavných susedných vrcholov akoby obal grafu, pozostávajúceho z vrstiev L_0 až L_i . Ak tieto vrcholy z vonkajšieho obalu odstránime, narazíme na ďalší obal z vrcholov M_1 . Postupným odstraňovaním vrstiev M_j odstránime všetky vrcholy a dostaneme sa až k vrcholom vrstvy L_i .

Z predchádzajúcej lemy vyplýva, že $|M_0| \geq 1$. Ľahko možno ukázať, že ak $N_k \neq \emptyset$ potom $|M_k| \geq 1$. To vyplýva z toho, že ak uvažujeme len „podgraf“ pozostávajúci z vrcholov N_k vo vrstve L_{i+1}, predošlá lema je rovnako dobre aplikovateľná, pretože tento graf spĺňa všetky predpoklady predchádzajúcich tvrdení. Ďalej indukciou možno ukázať, že ak $N_k = \emptyset$, tak $\forall j \geq k$ platí $M_j = \emptyset$. Nech m je najväčší taký index, že $M_m \neq \emptyset$. Je zrejmé, že množiny $M_0, \ldots, M_m$ sú disjunktné, neprázdne a platí $\bigcup_{k=0}^{m} M_k \subset L_{i+1}$. Teda máme, že $m \leq |L_{i+1}|$.

Keďže v polynomiálnom čase vieme pre ľubovoľný bod B v rovine a vrchol $w \in L_{i+1}$ určiť, či $B \in Area(w)$, vieme skonštruovať polynomiálny algoritmus, ktorý pre vrcholy z vrstvy L_{i+1} určí ich zaradenie do množín M_k, t.j. skonštruuje množiny $M_0, \ldots, M_m$. Pripomeňme, že pod vrstvou L_{i+1} rozumieme tie vrcholy, ktoré nám „ostali v hre“ po 1. kole vysielania správy, t.j. vrcholy s $|S(v)| \geq 2$.

Lema 5.3: *Pre každú z množín M_j, $0 \leq j \leq m$ a pre každý vrchol $v \in L_i$ platí, že vrchol v je hlavným susedným vrcholom pre nanajvýš dva vrcholy z množiny M_j.*

Dôkaz: Sporne predpokladajme, že vrchol $v \in L_i$ je hlavným susedným vrcholom pre aspoň 3 rôzne vrcholy z množiny M_j. Z týchto vrcholov si vyberme nejaké 3 vrcholy $a, b, c \in M_j$, pričom v danom nakreslení sú susedia vrcholu v po rade $P(v), a, c, b$. Z toho, ako je množina M_j definovaná máme, že žiaden z týchto prvkov $a, b, c \in M_j$ neleží v oblasti iného prvku z M_j. Zoberme si oblasti $Area(a)$ a $Area(b)$. Tieto oblasti majú spoločnú prinajmenšom hranu $(v, P(v))$, ktorá leží na hraniciach oboch oblastí a naviac je táto hrana aj hranicou oblasti $Area(c)$. Vrchol $c \notin Area(a) \cup Area(b)$ a teda aj hrana (c, v) s výnimkou bodu v leží mimo $Area(a) \cup Area(b)$. Zoberme si teraz hranicu oblasti $Area(c)$. Tá sa skladá z časti ležiacej mimo oblasť $Area(a) \cup Area(b)$ a z časti ležiacej v oblasti $Area(a) \cup Area(b)$. Keďže každá cesta z vrcholu a do vrcholu b vedie nutne cez hranicu oblasti $Area(c)$, musí v oblasti $Area(c)$ ležať aspoň jeden z vrcholov a alebo b, čo je spor s vlastnosťou prvkov množiny M_j. ♣



Obrázok 5.2: Ilustračný obrázok k dôkazu lemy 5.3

Ku každej množine M_j teraz definujme neorientovaný graf $G_j = (V_j, E_j)$ následovne:

$$V_j = M_j \cup \{v \in L_i \mid v \text{ je hlavný susedný vrchol nejakého vrcholu z } M_j\}$$

$$E_j = \{(v, w) \mid v \in M_j \wedge w \in V_j \cap L_i \text{ je hlavný susedný vrchol vrcholu } v\}$$

Takto definovaný graf $G_j = (V_j, E_j)$ je až na orientáciu hrán podgrafom grafu $G = (V, E)$. Pozrime sa teraz na štruktúru vzniknutých grafov. Graf $G_j = (V_j, E_j)$ pozostáva z vrcholov vrstvy L_{i+1}, ktoré patria do množiny M_j a ich hlavných susedných vrcholov (časť vrcholov vrstvy L_i). Každý vrchol v L_{i+1} má práve dva hlavné susedné vrcholy. Z predošlej lemy máme, že každý vrchol $v \in L_i$ je hlavným susedným vrcholom pre nanajvýš dva vrcholy z množiny M_j . Teda v grafe $G_j = (V_j, E_j)$ pre každý vrchol $v \in V_j$ platí, že $1 \le \deg_{G_j}(v) \le 2$. Je ľahko ukázať, že pre jednoduchý graf s touto vlastnosťou platí, že jeho komponenty súvislosti sú cesty a kružnice. Naviac graf $G_j = (V_j, E_j)$ je bipartitný a teda všetky kružnice sú párnej dĺžky, v ktorých sa striedajú hlavné susedné vrcholy a vrcholy z M_j. Každá cesta začína a končí nejakým hlavným susedným, pričom na ceste sa rovnako striedajú hlavné susedné vrcholy a vrcholy z množiny M_j (cesty majú rovnako párnu dĺžku).

Zavedieme si ďalší pojem, ktorý možno mierne zjednoduší zápis nasledujúcich tvrdení.

Definícia 5.2 (Kolízny graf): *Uvažujme komunikačnú rádiovú sieť popísanú grafom dosiahnuteľnosti $G = (V, E)$. Nech $V_1 \subseteq V$ je množina tých vrcholov grafu, ktoré už prijali zdrojovú správu a nech $V_2 = V - V_1$ sú vrcholy, ktoré ešte túto správu neprijali. Kolíznym grafom grafu $G = (V, E)$ a množiny vysielačov V_1 nazveme neorientovaný graf $G_K = (V_K, E_K)$ taký, že:*

$$V_K = V_1$$

$$E_K = \{(v, w) \mid v, w \in V_K \wedge \exists\, u \in V_2 \text{ taký, že } (v, u) \in E \wedge (w, u) \in E\}$$

Kolízny graf je graf tvorený vrcholmi, ktoré už prijali zdrojovú správu, pričom hrana v tomto grafe existuje medzi vrcholmi práve vtedy, ak existuje taký vrchol (medzi vrcholmi, ktoré ešte zdrojovú správu neprijali), že súčasné vysielanie oboch vrcholov by viedlo ku kolízii prijímania zdrojovej správy v tomto vrchole.

Lema 5.4: *Nech je kolízny graf $G_K = (V_K, E_K)$ vrcholovo α -zafarbiteľný, tak potom v grafe $G = (V, E)$ je možné vykonať bezkolízne vysielanie v α kolách, ktoré zabezpečí, že každý sused vrcholu z množiny V_1 prijme zdrojovú správu.*

Dôkaz: Bezkolízne vysielanie je určené tak, že vrchol $v \in V_1$ vysiela v kole *i*, ak jeho farba v α -zafarbení je *i*. Ukážeme, že takto definované vysielanie spĺňa tvrdenie lemy. Nech $w \in V_2$. Označme $N(w) = \{v \in V_1 \mid (v, w) \in E)$. Uvažujeme len tie $w \in V_2$, pre ktoré $|N(w)| \ge 1$. Je zrejmé, že množina vrcholov z $N(w)$ tvorí v kolíznom grafe kliku a teda každý z týchto vrcholov je zafarbený inou farbou. To znamená, že každý z týchto vrcholov bude vysielať v inom kole a teda vo vrchole w nenastane kolízia. ♣

Pozrime sa teraz na graf $G_0 = (V_0, E_0)$ a skonštruujme k nemu kolízny graf vzhľadom na množinu vrcholov, ktoré už správu prijali, t.j. vzhľadom na vrcholy z $V_0 - M_0 \subseteq L_i$. Tento kolízny graf bude pozostávať s komponentov súvislosti, ktoré budú samostatné vrcholy, cesty alebo kružnice. To vyplýva zo štruktúry grafu $G_0 = (V_0, E_0)$. Je jasné, že takýto kolízny graf je vrcholovo 3-zafarbiteľný, pričom takéto zafarbenie je algoritmicky ľahko nájsť. Priraďme teraz hlavným susedným vrcholom z množiny V_0 na základe určenej farby vysielaciu postupnosť. Konkrétne, každému vrcholu budeme priraďovať práve jedno z 3 kôl, v ktorom bude vysielať. Vrchol s priradenou farbou i $(1 \le i \le 3)$ vysiela v kole $1+i$. Takéto vysielanie zabezpečí doručenie zdrojovej správy do vrcholov z množiny M_0 (samozrejme za predpokladu, že vysielanie vedľajších susedných vrcholov toto vysielanie nepokazí).

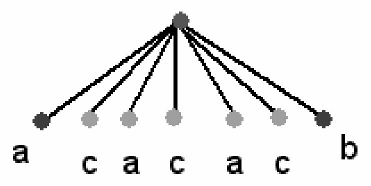
Rovnaké tvrdenie platí i pre ďalšie skonštruované grafy $G_j = (V_j, E_j)$, $j \ge 1$. Konkrétne platí, že k nim skonštruované kolízne grafy pozostávajú z komponentov súvislosti, ktoré sú kružnice, cesty alebo vrcholy. To vyplýva z definície kolízneho grafu a charakteru jednotlivých grafov $G_j = (V_j, E_j)$ - biparticita a vlastnosť, že $1 \le \deg_{G_j}(v) \le 2$.

Nasledujúcimi dvoma „procedúrami" ukážeme, ako ďalej (po priradení vysielacích kôl hlavným susedným vrcholom vrcholov z množiny M_0) priraďovať vrcholom z vrstvy L_i farby (vysielacie kolá) tak, aby sme zabezpečili úspešné doručenie správy do vrstvy L_{i+1}.

Procedúra Set-Minor-Neighbour

Predpokladajme, že pre susedné vrcholy (hlavné aj vedľajšie) vrcholov množín $M_0, \ldots, M_{j-1}$ a pre hlavné susedné vrcholy vrcholov množiny M_j ($j \ge 0$) máme skonštruované také priradenie vysielacích kôl (pre každý vrchol kolo 2, 3, alebo 4), ktoré zabezpečí doručenie zdrojovej správy do každého z vrcholov z množiny $\bigcup_{k=0}^{j} M_k$ a vedľajšie susedné vrcholy vrcholov z M_j ešte nemajú priradené žiadne vysielacie kolo. Potom vieme skonštruovať také priradenie vysielacích kôl pre vedľajšie susedné vrcholy vrcholov z M_j, ktoré neovplyvní úspešné doručenie správ do vrcholov z $\bigcup_{k=0}^{j} M_k$ a navyše má jednu užitočnú vlastnosť, ktorú využijeme v procedúre *Set-Major-Neighbour*.

Zoberme si vrchol $v \in M_j$, ktorý má aspoň jeden vedľajší susedný vrchol. Vzhľadom na nakreslenie, ktoré uvažujeme, očíslujme vrcholy z množiny $S(v)$ v nejakom fixnom smere (napríklad v smere hodinových ručičiek) tak, že jednému z hlavných susedných vrcholov bude priradené poradové číslo 1 a druhému z hlavných susedných vrcholov vrcholu v, bude priradené poradové číslo $|S(v)|$. Teda vedľajšie susedné vrcholy budú mať priradené čísla $2, \ldots, |S(v)|-1$. Keďže už skonštruované priradenie vysielacích kôl zabezpečuje doručenie správy do vrcholu v, musí mať tú vlastnosť, že dvom hlavným susedným vrcholom sú priradené rôzne vysielacie kolá. Ostalo nám ešte jedno vysielacie kolo. Nech prvému hlavnému susednému vrcholu (s poradovým číslom 1) je pridelené vysielacie kolo *a*, druhému *b* a nepoužité vysielacie kolo je *c*. Vedľajšiemu susednému vrcholu s párnym poradovým číslom priradíme vysielacie kolo *c* a vedľajšiemu susednému vrcholu s nepárnym poradovým číslom vysielacie kolo *a*. Pri takomto priradení dostane vrchol v zdrojovú správu určite aspoň vo vysielacom kole *b* (vrchol v má totiž iba jedného suseda, ktorý vysiela v kole *b*).



Obrázok 5.3: Ilustračný obrázok k procedúre Set-Minor-Neighbour

Dodajme, že vedľajší susedný vrchol vrcholu z množiny M_j je susedným vrcholom práve jedného vrcholu z množiny $\bigcup_{k=0}^{j} M_k$. Ak by bol totiž vrchol u, ktorý je vedľajší susedný vrchol vrcholu $v \in M_j$ a teda ležiaci vo vnútri oblasti $Area(v)$, susedným vrcholom nejakého iného vrcholu w z množiny $\bigcup_{k=0}^{j} M_k$, znamenalo by to, že existuje cesta dĺžky 1 medzi vrcholom u a vrcholom w. Keďže vrchol $v \in M_j$, žiaden vrchol z množiny $\bigcup_{k=0}^{j} M_k$ neleží vo vnútri oblasti $Area(v)$. Teda vrchol w leží mimo oblasť $Area(v)$. Každá cesta z vrcholu w do vrcholu u musí prejsť cez hranicu oblasti $Area(v)$ a to vďaka planarite cez nejaký vrchol na hranici oblasti $Area(v)$. Teda každá cesta musí mať dĺžku aspoň 2, čo je spor s tým, že existuje cesta dĺžky 1 medzi vrcholom u a vrcholom w. ♣

Procedúra Set-Major-Neighbour

Predpokladajme, že pre susedné vrcholy (hlavné aj vedľajšie) vrcholov množín $M_0, \ldots, M_{j-1}$ ($j \geq 1$) máme skonštruované také priradenie vysielacích kôl procedúrami *Set-Major-Neighbour* a *Set-Minor-Neighbour* (pre každý vrchol kolo 2, 3 alebo 4), ktoré zabezpečí doručenie zdrojovej správy do každého z vrcholov z množiny $\bigcup_{k=0}^{j-1} M_k$. Zoberme teraz množinu vrcholov $M_j \subseteq L_{i+1}$. Skonštruujeme teraz také priradenie vysielacích kôl hlavným susedným vrcholom vrcholov z množiny M_j, ktoré nebude v konflikte ani s už priradenými vysielacími kolami a ani neovplyvní doručenie zdrojovej správy do vrcholov z množiny $\bigcup_{k=0}^{j-1} M_k$. Naviac zabezpečí úspešné doručenie zdrojovej správy do všetkých vrcholov množiny M_j. K množine M_j máme priradený graf $G_j = (V_j, E_j)$. Na zabezpečenie úspešného doručenia zdrojovej správy do vrcholov z M_j opäť potrebujeme zabezpečiť 3-zafarbiteľnosť kolízneho grafu pre graf $G_j = (V_j, E_j)$. Oproti riešeniu v prípade grafu $G_0 = (V_0, E_0)$ tu máme naviac ten problém, že niektoré hlavné susedné vrcholy vrcholov z M_j už možno majú priradené vysielacie kolo, pretože sú susednými vrcholmi vrcholov z $\bigcup_{k=0}^{j-1} M_k$. Teda potrebujeme 3-zafarbiteľnosť kolízneho grafu v prípade, že niektoré z hlavných susedných vrcholov už majú predchádzajúcim behom algoritmu určenú farbu. Pozrime sa na komponenty súvislosti kolízneho grafu. Ak v nejakom komponente nie je žiadna farba dopredu určená, je nájdenie 3-zafarbenia jednoduchá vec. Komponent kolízneho grafu, kde už nejaká farba určená je, môžeme vzhľadom na jeho štruktúru rozdeliť na cesty, kde má farbu (vysielacie kolo) priradený len začiatočný vrchol cesty, a na cesty, kde majú farbu priradené obe koncové vrcholy cesty. Takto vytvorené cesty sú až na vrcholy s určenými farbami disjunktné. V prípade, že farba je určená len pre počiatočný vrchol cesty, máme k dispozícii 2 voľné farby, ktorými vieme striedavo vyfarbiť zvyšok cesty.

Uvažujme prípad, že farba je určená pre obe konce cesty. Ak sú tieto farby rôzne (nech sú to farby *a*, *b*) , máme voľnú ešte jednu farbu *c* a cesty vyfarbime tak, ako sme to robili v prípade procedúry *Set-Minor-Neighbour*. Teda vyfarbíme cestu postupne farbami *a, c, a, c, a, c, ...,* *b*. V prípade, že obe konce cesty sú ofarbené rovnakou farbou (nech je to *a*) a cesta má dĺžku aspoň 2 (t.j. má aspoň jeden neofarbený vnútorný vrchol) použijeme nasledovné ofarbenie cesty: *a, b, c, b, c, b, c, ..., a* , kde *b, c* sú nepoužité farby. Ak cesta má dĺžku 1 (t.j. cestou je hrana, kde s ňou inicidentné vrcholy sú ofarbené rovnakou farbou), máme kolíziu v ofarbení, ktorú nevieme vyriešiť. Takáto situácia ale vďaka ofarbeniam použitým v procedúre *Set-Minor-Neighbour* a procedúre *Set-Minor-Neighbour* nikdy nenastane. Z definície grafu $G_j = (V_j, E_j)$ a jeho kolízneho grafu vyplýva, že ak v kolíznom grafe existuje hrana, kde obe vrcholy majú rovnakú farbu, tak v grafe $G_j = (V_j, E_j)$ existuje vrchol $v \in M_j$, ktorého hlavné susedné vrcholy majú dopredu určenú rovnakú farbu (vysielacie kolo). Keďže $j \geq 1$ existuje vrchol $w \in M_{j-1}$ taký, že $Area(v) \subset Area(w)$. Obom hlavným susedným vrcholom vrcholu v bolo už priradené vysielacie kolo. Z toho vyplýva, že hlavný susedia vrcholu v sú zároveň aj nejaký susedia vrcholu w, pretože jediné vrcholy v oblasti $Area(w)$, ktoré majú priradené vysielacie kolo, sú nutne susedia vrcholu w. Ďalej vo vnútri oblasti $Area(v)$ nemôže ležať žiaden zo susedných vrcholov vrcholu w (požadovaná cesta dĺžky 1 nemôže prejsť hranicu oblasti $Area(v)$). Opäť ako v procedúre *Set-Minor-Neighbour,* vzhľadom na nakreslenie, ktoré uvažujeme, očíslujme vrcholy z množiny $S(w)$ v nejakom fixnom smere (hodinových ručičiek) tak, že jednému z hlavných susedných vrcholov bude priradené poradové číslo 1, druhému z hlavných susedných vrcholov vrcholu w, bude priradené poradové číslo $|S(w)|$ a vedľajšie susedné vrcholy budú mať priradené čísla $2, \ldots, |S(w)| - 1$. Zoberme si teraz dva susedné vrcholy vrcholu w s nasledujúcimi poradovými číslami: $u_k, u_{k+1} \in S(w)$ a uvažujme oblasť $Area(w, (u_k, u_{k+1}))$. Možno ukázať, že vo vnútri tejto oblasti neleží žiaden zo susedov vrcholu w. Keďže hlavné susedné vrcholy vrcholu v sú susedia vrcholu w, majú v zmysle uvedeného očíslovania nejaké poradové čísla. Tieto čísla musia byť nasledujúce. Ak by ich nemali nasledujúce, vedeli by sme skonštruovať oblasť, vo vnútri ktorej leží nejaký sused $u \in L_i$ vrcholu w, z ktorého ale nevedie cesta žiadna zdrojová cesta. Všimnime si, že procedúra *Set-Minor-Neighbour* takýmto susedom s nasledujúcimi poradovými číslami priradí vždy rôzne farby (vysielacie kolá). Ak teda použijeme na prideľovanie vysielacích kôl pre vedľajšie susedné vrcholy procedúru *Set-Minor-Neighbour*, tak spomínaná konfliktná situácia nikdy nenastane. ♣

Samotný algoritmus, ktorý pridelí každému vrcholu vrstvy L_i vysielacie kolá funguje tak, že najprv vykoná opísaným priamym spôsobom pridelenie vysielacích kôl hlavným susedným vrcholom vrcholov z množiny M_0. Na pridelenie vysielacích kôl pre vedľajšie susedné vrcholy vrcholov z M_0 sa použije procedúra *Set-Minor-Neighbour*. Nech m je najväčší taký index, že $M_m \neq \emptyset$. Potom nasleduje $2m$ fáz algoritmu, pričom vo fáze $2j-1$ sa aplikuje procedúra *Set-Major-Neighbour* na množinu M_j a vo fáze $2j$ sa aplikuje procedúra *Set-Minor-Neighbour* na množinu M_j.

Z fungovania jednotlivých procedúr vyplýva, že aplikovaním predchádzajúceho algoritmu získame také pridelenie vysielacích kôl (2, 3, alebo 4) niektorým vrcholom z vrstvy L_i, ktoré zabezpečí doručenie zdrojovej správy do každého vrcholu v z vrstvy L_{i+1} takého, že $|S(v)| \geq 2$. Vrcholy, ktorým nebolo pridelené žiadne vysielacie kolo, počas kôl 2, 3 a 4 nevysielajú. Vrchol v z L_{i+1}, pre ktorý platí, že $|S(v)| = 1$ prijíma zdrojovú správu v kole 1, kedy vysiela každý vrchol z vrstvy L_i.

Opäť poznamenajme, že pridelenie vysielacích kôl vieme vykonať v polynomiálnom čase, pretože každú z použitých elementárnych procedúr vieme vykonať v polynomiálnom čase a každá sa vykonáva polynomiálne veľa krát.

Na základe uvedeného algoritmu vieme u planárnych grafov dosiahnuteľnosti vygenerovať prijateľný rozvrh, ktorý vykoná broadcasting v čase $4.D$. Uvedený algoritmus priradenia vysielacích kôl má tú vlastnosť, že každý vrchol, ktorý má takého suseda v vo vrstve L_{i+1}, že platí $|S(v)| \geq 2$, bude vysielať v jednom z kôl 2, 3 alebo 4. Toto vysielanie ale doručí správu aj susedným vrcholom s $|S(v)| = 1$. Tým je uvedené 1. kolo vysielania pôvodnej schémy pre tento vrchol zbytočné. Na druhej strane, ak vrchol $w \in L_i$ nemá žiadneho suseda $v \in L_{i+1}$ s vlastnosťou $|S(v)| \geq 2$, algoritmus mu nepriradí žiadne vysielacie kolo. Zároveň však vysielanie vrcholu w nemôže interferovať s vysielaním vrcholov, ktoré vysielajú v 2., 3. alebo 4. kole pôvodného rozvrhu a teda jeho vysielanie, ktoré bolo pôvodne v 1. kole, môže byť vykonané v ľubovoľnom z kôl 2., 3. alebo 4. Ako vidieť, vysielanie zdrojovej správy z vrstvy L_i do vrstvy L_{i+1} vieme naplánovať v 3 kolách, čo pre celkovú dĺžku vygenerovaného rozvrhu dáva $3.D$ potrebných kôl.

Uvedený mechanizmus konštrukcie prijateľného rozvrhu rádiového broadcastingu funguje s menšími úpravami aj pre orientované planárne grafy dosiahnuteľnosti. Modifikácia je tá, že vždy uvažujeme iba orientované hrany z vrstvy L_i do vrstvy L_{i+1}. Samozrejme aj rozdelenie vrcholov do vrstiev je určené na základe dĺžky najkratšej orientovanej cesty zo zdrojového vrcholu. Keďže po orientovaných hranách nepotrebujeme vysielať z vrstvy L_{i+1} do vrstvy L_i, orientácia týchto hrán nerobí problém. Ak existuje nejaká orientovaná hrana z vrstvy L_j do vrstvy L_i, kde $j \geq i$, tak tá nám rovnako neprekáža, pretože postupujeme po vrstvách a potenciálne budúce vysielanie na tejto hrane bude viesť len do vrcholu, ktorý už zdrojovú správu úspešne prijal a v danom kole už žiadne vysielania nevykonáva ani neprijíma.

Algoritmus opísaný v predchádzajúcom texte môžeme zhrnúť do nasledujúceho tvrdenia:

Tvrdenie 5.3: *Nech $G = (V, E)$ je orientovaný planárny graf dosiahnuteľnosti rádiovej siete. Potom existuje deterministický algoritmus, ktorý v polynomiálnom čase pre daný zdrojový vrchol s vygeneruje prijateľný rozvrh Π rádiového broadcastingu taký, že* $|\Pi| = 3.D$.

Kapitola 6

Záver

V tejto práci, venovanej predovšetkým otázke broadcastingu v rádiových sieťach popísaných grafom dosiahnuteľnosti (grafový model rádiových sietí), sme sa snažili splniť dve základné ciele:

- podať prehľad aktuálnych výsledkov dosiahnutých pri riešení úlohy broadcastingu v modeloch s rôznymi komunikačnými scenármi spolu s opisom najzaujímavejších a najpoužívanejších techník riešenia
- navrhnúť a analyzovať efektívne broadcastovacie algoritmy pre rádiové siete s obmedzenou topologickou štruktúrou grafu dosiahnuteľnosti

Ako vidieť v prehľade aktuálnych výsledkov, ktorému je venovaná najmä 3. kapitola tejto práce, je oblasť rádiových sietí v poslednom období intenzívne skúmaná. Výsledkom tohto výskumu je množstvo nových a efektívnejších algoritmov s prepracovanejšími technikami pri využití rôznych predpokladov kladených na komunikačné procesy. Úroveň informácie o sieti vo vrcholoch siete a predpoklady kladené na komunikačné procesy majú najväčší vplyv na celkovú efektívnosť riešenia komunikačných úloh v rádiových sieťach. Preto je pri reálnych aplikáciách dôležité vybrať model, ktorý čo najlepšie charakterizuje sieť. Nové výsledky postupne znižujú medzeru medzi hornými a dolnými odhadmi času realizácie komunikačných úloh – ako hlavného merítka efektívnosti jednotlivých riešení. I keď je táto oblasť intenzívne skúmaná, stále existuje množstvo otvorených problémov, na ktoré sa nepodarilo nájsť odpoveď. Pre väčšinu najpoužívanejších modelov ešte stále nie sú známe asymptoticky optimálne algoritmy. Najširší priestor je však v tejto práci venovaný návrhu a analýze broadcastovacích algoritmov pre siete s obmedzenou topologickou štruktúrou grafu dosiahnuteľnosti.

V kapitole 3.3 tejto práce prezentujeme sériu algoritmov riešiacich problém distribuovaného priradenia čísla vrstvy vrcholu a nájdenia odhadu najväčšej hodnoty uloženej vo vrcholoch siete v modely s distribuovaným riadením, s detekciou kolízie a bez využitia spontánnych vysielaní, kde jediná dostupná informácie vo vrcholoch je ich identifikátor. S využitím týchto algoritmov sa podarilo skonštruovať v tomto modely broadcastovací algoritmus pre všeobecné symetrické rádiové siete realizujúci broadcasting v čase $O(n + D\log D)$ kôl. Zároveň technika použitá v tomto algoritme ukazuje možnosť, ako každý algoritmus využívajúci spontánne vysielania a pracujúci v čase $T(n)$ kôl v modely s distribuovaným riadením a detekciou kolízie transformovať na algoritmus nevyužívajúci spontánne vysielania a pracujúci v čase $T(n) + O(D\log D)$ kôl.

V 4. kapitole sme sa zaoberali otázkou broadcastingu v rádiových sieťach, ktorých grafom dosiahnuteľnosti je dvojrozmerná symetrická mriežka. Navrhli sme sériu rôznych broadcastovacích algoritmov určených pre rôzne komunikačné scenáre, ktoré prezentovali rôzne prístupy k riešeniu tohto problému. Nakoniec pre najvšeobecnejší model – model s distribuovaným riadením bez detekcie kolízie, kde jediná informácie dostupná vo vrcholoch je ich identifikátor, sme skonštruovali broadcastovací algoritmus pracujúci asymptoticky optimálne v čase $O(D + \log n) \subseteq O(D)$ kôl. Využili sme tu novú techniku vysielania riadeného vysielacím reťazcom, ktorý je priradený identifikátoru vrcholu.

5. kapitola tejto práce je venovaná centralizovanému broadcastingu v sieťach, ktorých graf dosiahnuteľnosti je planárny. Skonštruovali sme algoritmus generujúci prijateľné rozvrhy rádiového broadcastingu dĺžky $3.D$ pre planárne orientované grafy. Tento algoritmus využíva pomerne netradičný prístup. Namiesto štrukturálnych informácii, ktoré sú známe pre planárne grafy, sa algoritmus opiera o to, čo je planárnym grafom najvlastnejšie – existenciu planárneho nakreslenia. Zároveň algoritmus pracuje po vrstvách, čo v symetrických sieťach umožňuje spustenie jeho viacerých paralelne bežiacich inštancií („pipelining"). Tieto vlastnosti nám umožňujú predpokladať, že tento algoritmus by mohol byť využitý aj na konštrukciu rýchlych centralizovaných broadcastovacích algoritmov pre neplánarne symetrické grafy. Konkrétne pre grafy, ktoré síce nie sú planárne, no nejakým vhodným spôsobom sa dajú rozložiť na planárne grafy tak, aby postupné spúšťanie broadcastovacieho algoritmu v týchto planárnych grafoch viedlo k realizácii broadcastingu v celom grafe.

Táto práca poskytuje pohľad iba na malú časť širokej problematiky efektívneho riešenia komunikačných úloh v rádiových sieťach. Rádiové siete totiž umožňujú vysoký stupeň variácie v používaných modeloch vďaka možným kombináciám v predpokladoch kladených na komunikačné procesy. Mnohé oblasti tejto problematiky ostávajú stále nepreskúmané.

Literatúra

[ABLP91] N. ALON, A. BAR-NOY, N. LINIAL A D. PELEG, *A lower bound for radio broadcast*, Journal of Computer and System Sciences 43 (1991), 290-298

[BGI92] R. BAR-YEHUDA, O. GOLDREICH A A. ITAI, *On the time complexity of broadcast in radio networks: An exponential gap between determinism and randomization*, Journal of Computer and System Sciences 45 (1992), 104-126

[CGGPR00] B.S. CHLEBUS, L. GASIENIEC, A. GIBBONS, A. PELC A W. RYTTER, *Deterministic broadcasting in unknown radio networks*, Proceedings 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2000), 861-870, 2000

[CGR00] M. CHROBAK, L. GASIENIEC A W. RYTTER, *Fast broadcasting and gossiping in radio networks*, Proc. 41st Symposium on Foundations of Computer Science (FOCS'2000), 575-581

[CK85] I. CHLAMTAC A S. KUTTEN, *On broadcasting in radio networks – problem analysis and protocol design*, IEEE Transactions on Communications 33 (1985), 1240 – 1246

[CMS01] A.E.F. CLEMENTI, A. MONTI A R. SILVESTRI, *Selective families, superimposed codes and broadcasting on unknown radio networks*, Proc. 12th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'2001), 709-718

[CR03] A. CZUMAJ A W. RYTTER, *Broadcasting algorithms in radio networks with unknown topology*, Proc. 44th Ann. Symposium on Foundations of Computer Science (FOCS'2003)

[CW87] I. CHLAMTAC A O. WEINSTEIN, *The Wave Expansion Approach to Broadcasting in Multihop Radio Networks*, Proc. INFOCOM, 1987

[DP01] A. DESSMARK A A.PELC, *Tradeoffs between knowledge and time of communication in geometric radio networks*, Proc. 13th Ann. ACM Syposium on Parallel Algorithms and Architectures (SPAA 2001), 59-66

[EKa04] M. ELKIN A G. KORTSARZ, *A logarithmic lower bound for radio broadcast*. Journal of Algorithms 52 (2004), 8-25

[EKb04] M. ELKIN A G. KORTSARZ, *Polylogarithmic additive inapproximability of the radio broadcast problem*. Proc. 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'2004)

[EKc04] M. ELKIN A G. KORTSARZ, *An Improved algorithm for radio broadcast*, nepublikované, 2004

[GM95] I. GABER A Y. MANSOUR, *Broadcast in radio networks*, Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'1995), 577-585

[KP02] D. KOWALSKI A A. PELC, *Deterministic broadcasting time in radio networks of unknown topology*, Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'2002), 63-72

[KPa03] D. KOWALSKI A A. PELC, *Faster deterministic broadcasting in ad hoc radio networks*, Proc. 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'2003), 2003, Berlin, Nemecko, LNCS 2607, 109-120

[KPa04] D. KOWALSKI A A. PELC, *Time of deterministic broadcasting in radio networks with local knowledge*, SIAM Journal on Computing 33 (2004), 870-891.

[KPb03] D. KOWALSKI A A. PELC, *Broadcasting in undirected ad hoc radio networks*, Proc. 22nd Ann. ACM Symposium on Principles of Distributed Computing (PODC'2003), Júl 2003, Boston, USA, 73-82

[KPb04] D. KOWALSKI, A. PELC, *Centralized deterministic broadcasting in undirected multi-hop radio networks*, Proc. 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'2004), August 2004, Harvard University, Cambridge, U.S.A., LNCS 3122, 171-182.

[P02] A. PELC, *Broadcasting in radio networks*, in: I. Stojmenovic(Ed.), Handbook of Wireless Networks and Mobile Computing,John Wiley, New York, 2002

[S90] W. SCHNYDER, *Embedding planar graphs on the grid*. In Proc. 1st ACM-SIAM Symp. Discrete Algorithms, pp. 138-148, 1990.