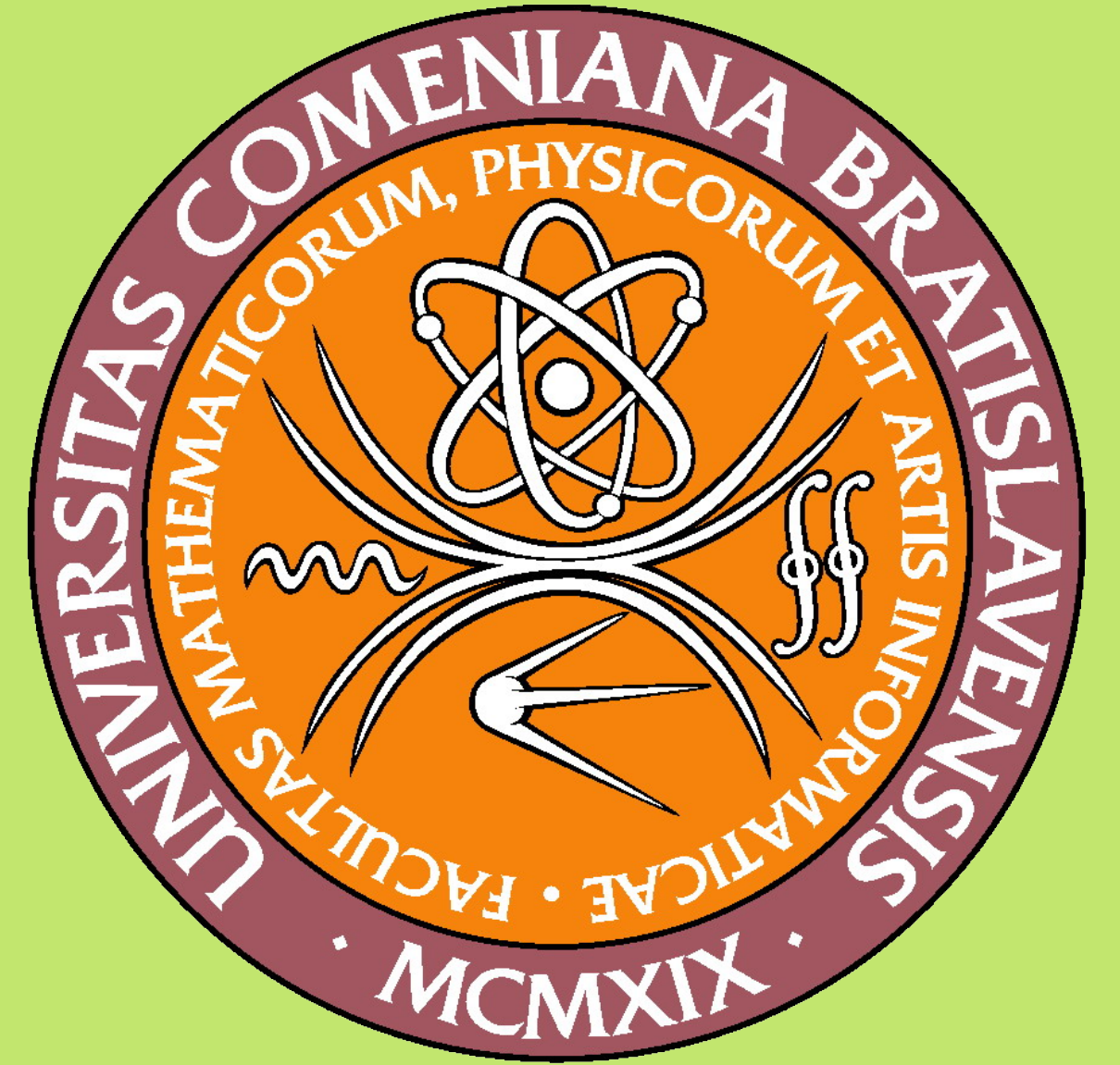


ViDA: Vizualizácia distribuovaných algoritmov

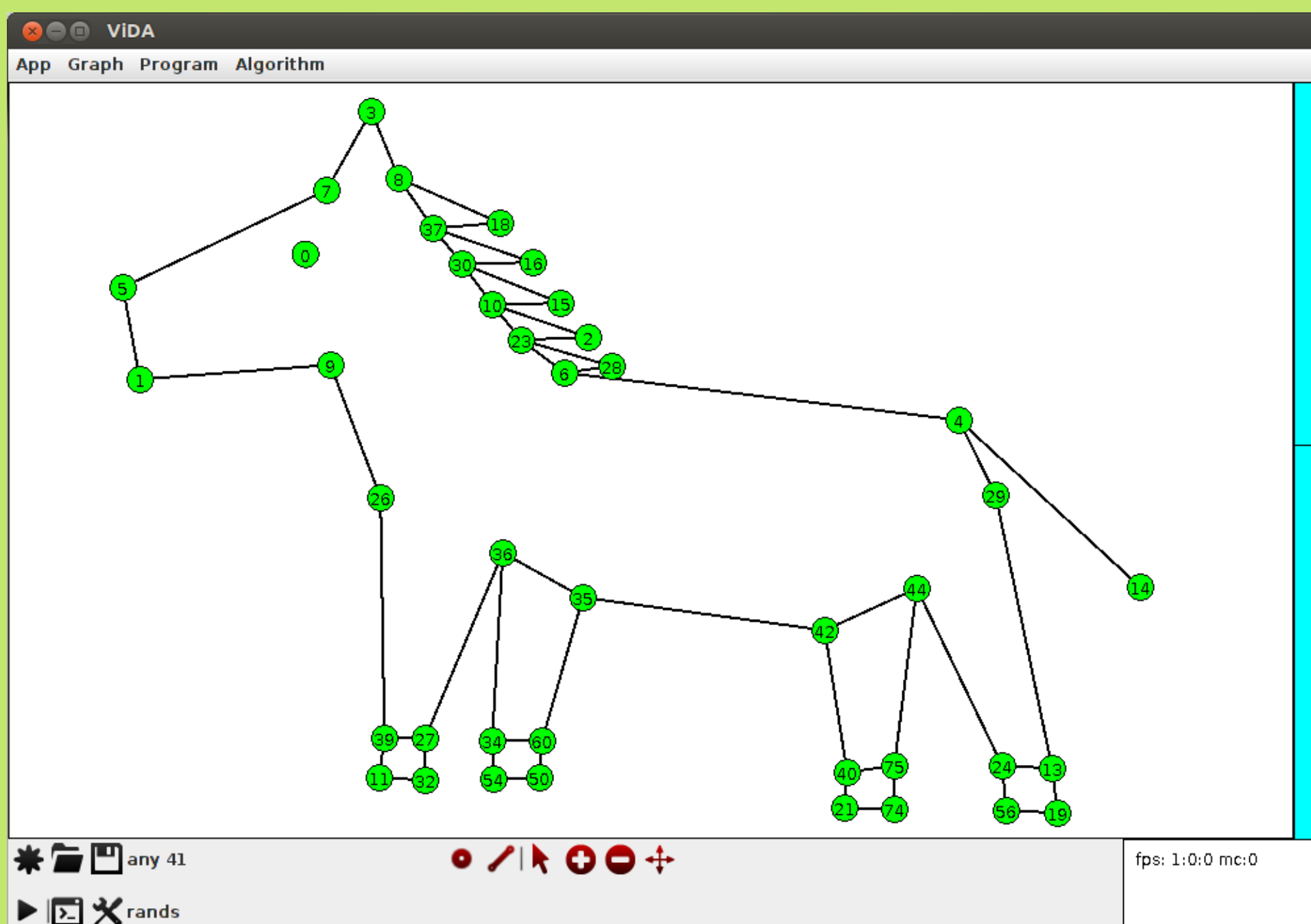
Michal Anderle¹, Ján Hozza¹
Supervisor: Jakub Kováč¹

¹ Katedra informatiky FMFI UK, Mlynská Dolina, 842 48 Bratislava



Úvod

V našom ročníkovom projekte sme sa rozhodli zaoberať vizualizáciou distribuovaných algoritmov pomocou Java aplikácie, ktorá umožňuje jednoduché a rýchle pochopenie témy, bez študovania dlhých odborných textov.



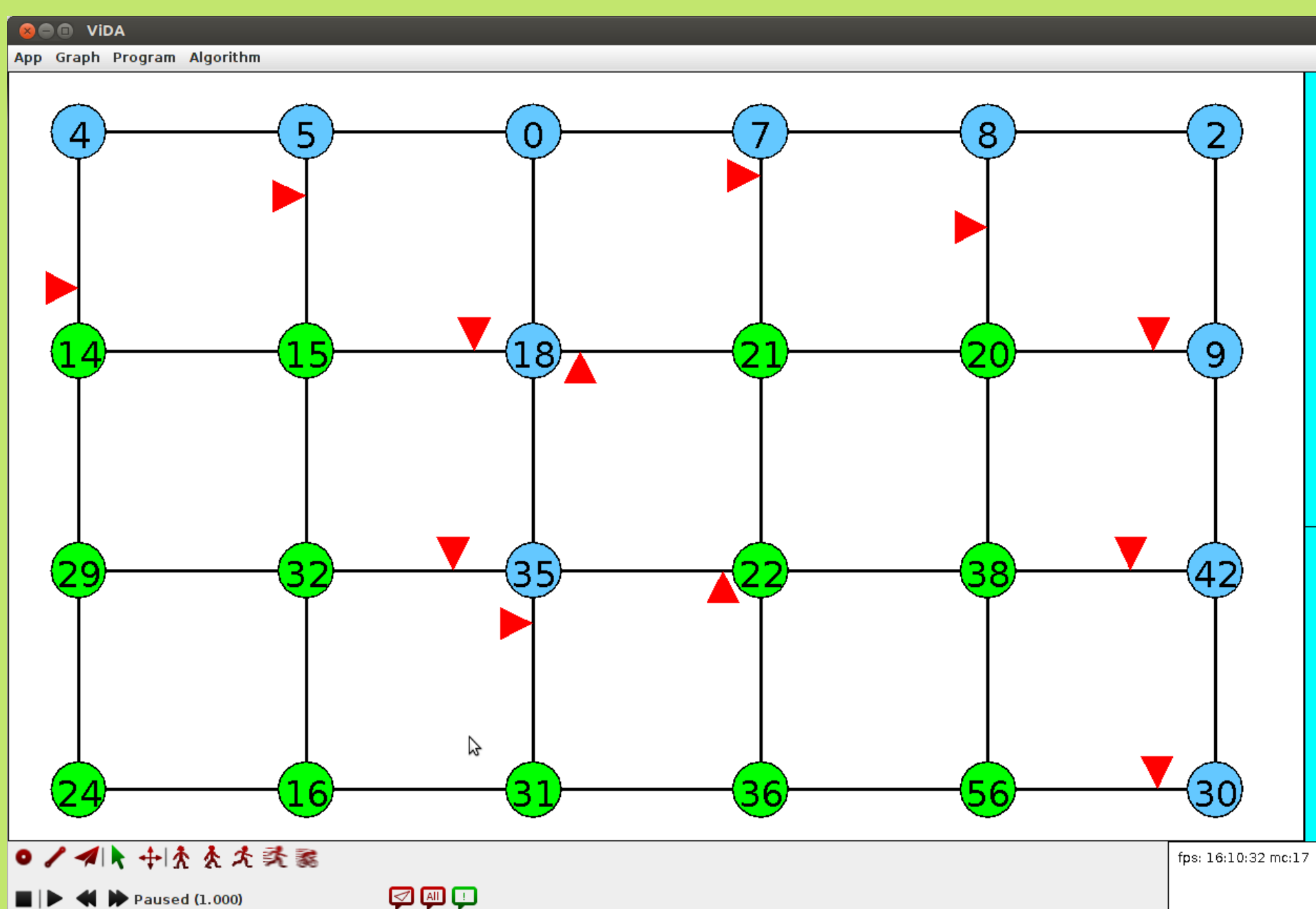
DISTRIBUOVANÉ ALGORITMY

Vlastnosti modelu:

- niekoľko počítačov zapojených do siete obojsmernými linkami
- majú jednoznačné id, komunikujú len správami
- správy sa nestrácajú, nemenia poradie, ale môže im to trvať ľubovoľne dlho – asynchrónna komunikácia

Ciele:

- poslať čo najmenej správ
- minimalizovať dobu behu algoritmu



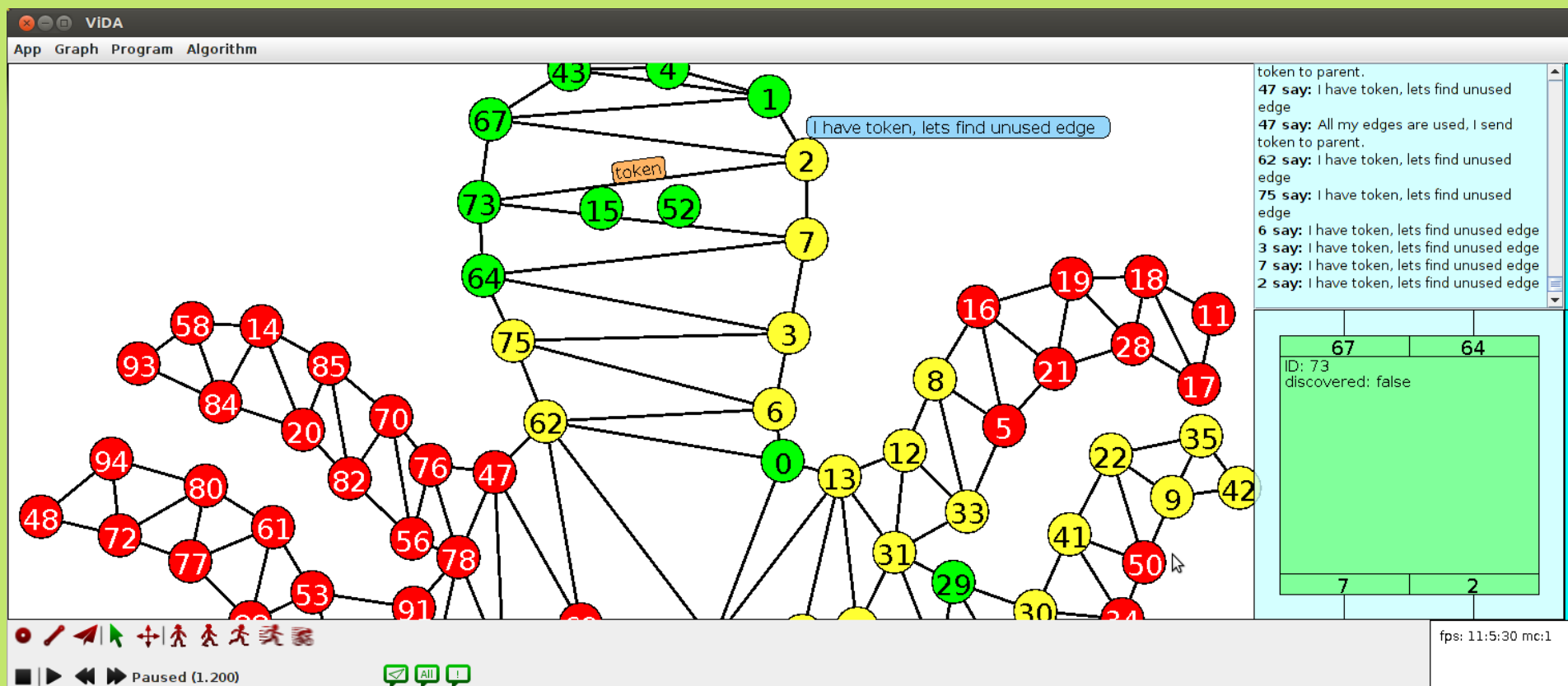
Obr. 2: Asynchrónna komunikácia – klebeta začala vľavo hore (vrchol 4) a dostala sa na druhý koniec mriežky (vrchol 30) skôr, než do spodného suseda (vrchol 14). Aj toto naša vizualizácia dokáže spraviť.

HLAVNÉ CIELE

- vizualizácie ušité na mieru konkrétnym distribuovaným algoritmom
- interaktivita s používateľom
- prehľadnosť a jednoduchosť používania aplikácie
- schopnosť vizualizovať vlastné algoritmy

VIZUALIZOVANÉ ALGORITMY

- broadcast – ako povedať novú klebetu všetkým v sieti?
- voľba šéfa na úplnom grafe – ako sa spomedzi niekoľkých identických programov dá zvoliť jeden šéf? a čo ak pri tom chceme poslať čo najmenej správ?
- traverzovanie – ako prehľadať celý graf podávaním si jediného tokenu?



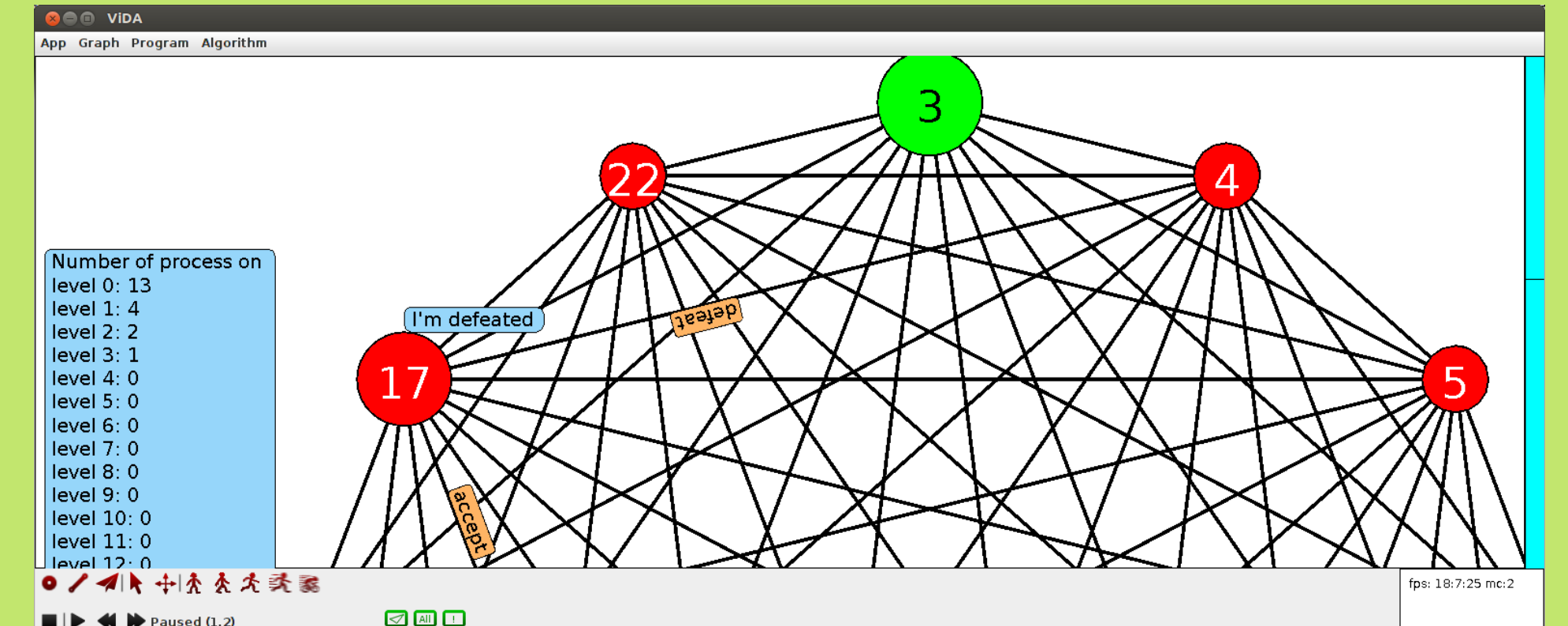
Obr. 3: Traverzovanie – zelené vrcholy sú nenavštívené, oranžové sú navštívené, červené sú úplne vybavené, teda už preskúmali všetkých susedov

NÁSTROJE

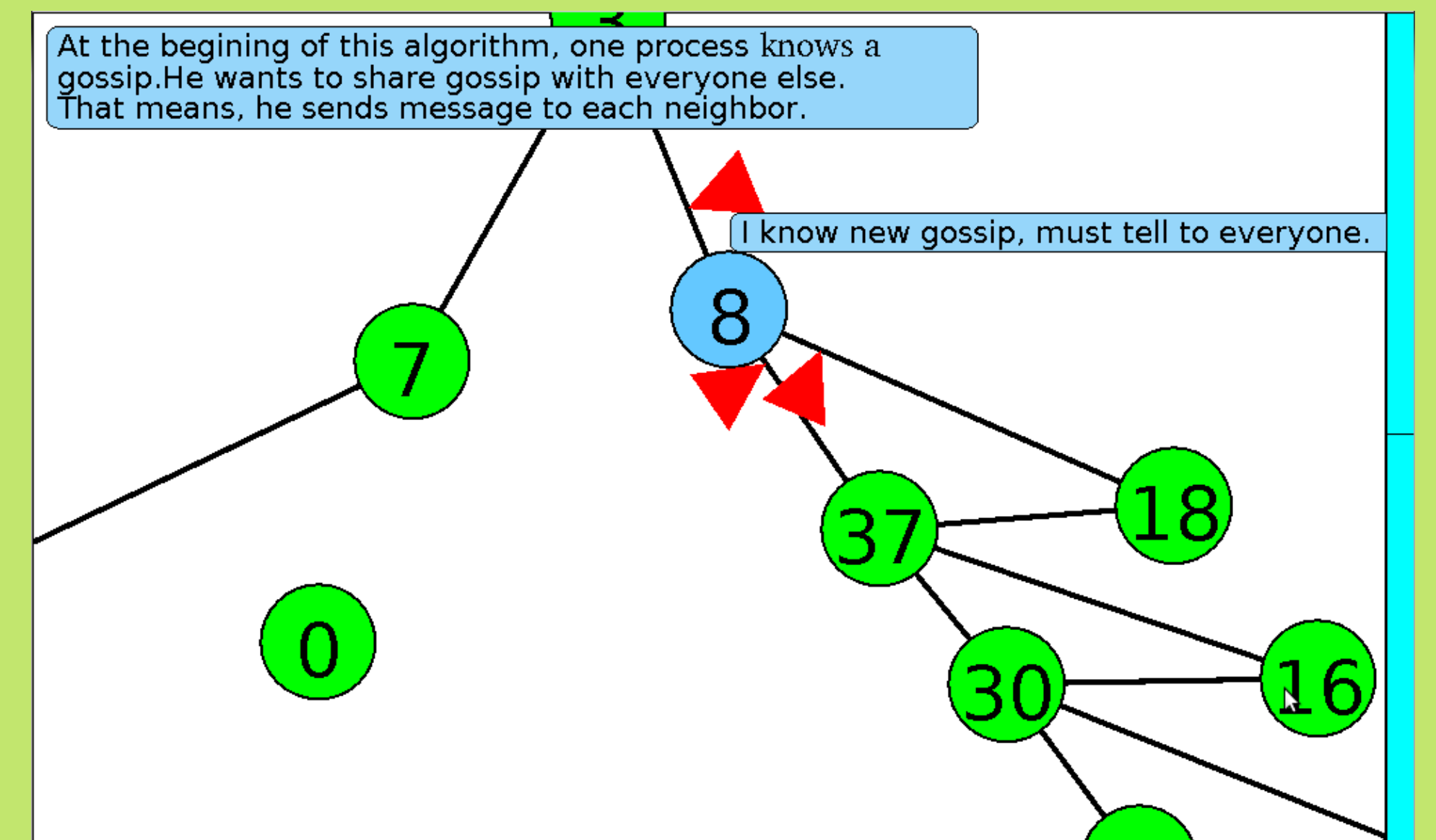
- úprava grafu
 - pridávanie, mazanie, editovanie, hýbanie vrcholov a hrán
 - ukladanie, automatické generovanie rôznych typov a veľkostí
 - verifikácia – napr. niektoré algoritmy sú určené len pre úplné grafy
- časovanie správ
 - možnosť chytiť správu a presunúť ju na iné miesto na hrane
 - nastavovanie rýchlostí hrán a správ
- selekcia – zobrazovanie dodatočných informácií

SPÔSOB VIZUALIZÁCIE

- zobrazovanie udalostí priamo v grafe – netreba vrtieť hlavou a hľadať, čo sa kde deje, informácie sa zobrazujú tam, kde sa ich to týka, všetko pomocou *vyskakovacích bubliniek*
- interaktivita – čo by sa stalo, keby...? užívateľ môže priamo ovplyvňovať, čo sa stane
- detekcia a vysvetlenie zaujímavých udalostí – keď sa niečo stane, aplikácia sa pozastaví a vysvetlí, čo sa stalo? prečo sa to stalo? kde sa to stalo? čo sa bude diať ďalej?



Obr. 4: Informácie priamo v grafe – intuitívne spojenie vizuálnych a hodnotových vlastností, napr. veľkosť vrcholu = level, farba vrcholu = stav (napr. červený = mŕtvy/porazený/neaktívny). Algoritmus na voľbu šéfa



Obr. 5: Zobrazovanie informácií v bublinkách. Algoritmus broadcast.

VLASTNÉ VIZUALIZÁCIE

- možnosť vytvárania vlastných vizualizácií
- knižnica *vidalib*
 - jednoduchá knižnica v C++ komunikujúca s našim programom
 - poskytnutie celej palety nástrojov – vyskakovacie bublinky, zmena farby a veľkosti vrcholov...
- hlbšie pochopenie algoritmu po jeho naimplementovaní
- potešenie z vlastných fungujúcich algoritmov

PLÁNY DO BUDÚCNOSTI

- ďalšie algoritmy
 - GHS – voľba šéfa na všeobecnom grafe
 - KKM – voľba šéfa s využitím traverzovania
 - routing – smerovanie dát v sieti. Kam poslať paket, aby sa dostal do cieľa?
 - problém dohody
- nové nástroje na prácu s grafom, správami, ďalšie nástroje na vizualizáciu
- viac zábavy, viac interaktivity – užívateľ sa môže zahrať na zákeráka a snažiť sa donútiť algoritmus, aby poslal čo najviac správ
- podpora viacerých programovacích jazykov