

Simulacija širenja epidemija u modernom društvu

Miloš Stanojević

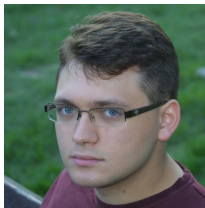
NEDELJA INFORMATIKE v2.5

14. april 2016.

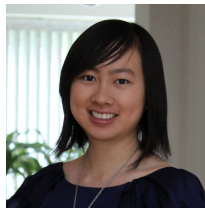
O nama



Petar Veličković



Andrej Ivašković



Stella Lau



Miloš Stanojević

Hack Cambridge



- ▶ Timsko 24-časovno takmičenje u razvijanju aplikacija (*hackathon*), pod pokroviteljstvom Univerziteta u Kembridžu i *Major League Hacking* organizacije.
 - ▶ Sponzorisano od strane velikog broja firmi (*Thales, GSA Capital, Bloomberg, Morgan Stanley, Palantir, Jane Street, ARM, Amazon, Google, Merrill Lynch...*),
- ▶ Učestvovalo oko 100 timova sa najvećih svetskih univerziteta (*Cambridge, MIT, Oxford, Imperial, ETH...*)
- ▶ *Viral* je proglašen za jedan od najboljih sedam projekata.
- ▶ Više informacija na <https://www.hackcambridge.com>.

Hack Cambridge



Motivacija



- ▶ Tradicionalno, modeliranje širenja epidemija se radi korišćenjem *jednoslojnih mreža*.
- ▶ U ovim mrežama, članovi populacije predstavljaju *čvorove* u grafu, a zaraza se može širiti preko njegovih *ivica*.
- ▶ Često korišćen *SIR (susceptible-infected-recovered)* model i njemu srodni modeli, u kome svakom čvoru pripisujemo trenutno *stanje*.

Nedostaci osnovnog pristupa



- ▶ Glavni nedostatak pristupa problemu koristeći jednoslojne mreže je da je *previše "kratkovid"*.
- ▶ U realnosti, nad populacijom može konkurentno da se širi *veliki broj procesa*, i ti procesi mogu *uticati* jedni na druge.
 - ▶ Ovo efektivno formira *mrežu mreža*!
- ▶ Multipleks mreže (o kojima će biti reči uskoro) su popularan način za predstavljanje ovih interakcija.

Primeri konkurentnih procesa



- ▶ Zadržavajući se unutar okvira jedne konkretne bolesti, možemo dobiti preciznije rezultate razmatrajući paralelno *različite puteve širenja te bolesti...*
 - ▶ ...npr. vazduhom, fizičkim kontaktom, ...
 - ▶ Svaki od ovih puteva predstavlja jedan odvojen sloj u mreži!
- ▶ Možemo se, zatim, proširiti na posmatranje *paralelnog širenja više zaraza*—time modelirajući njihovu *komorbidnost*.
 - ▶ Ponovo, svaka zaraza dobija poseban sloj u mreži.
- ▶ Konačno, možemo razmatrati jedan izuzetno važan šireći proces (pogotovo u modernom društvu) od koga zavisi *efektivnost prevencije i imunizacije...*

Informisanost



- ▶ Proces o kome je reč je širenje *informisanosti* o izbijanju epidemije unutar društva.
- ▶ Osobe koje su informisane o epidemiji mogu da sprovedu *preventivne mere* da bi smanjili verovatnoću infekcije (npr. ostajanje kod kuće, zaštitne maske, vakcinisanje...).
- ▶ U vreme nastanka SIR modela (*Kermack & McKendrick, 1927*) ovaj proces je mogao da se ignoriše (jer je širenje informisanosti u većini slučajeva zahtevalo fizičku interakciju).
- ▶ Međutim, u modernom društvu, razvojem tehnologije (pogotovo *društvenih mreža*) je moguće širiti informacije bez ikakvog fizičkog kontakta!
 - ▶ Očigledno, informisanost mora da dobije svoj sloj...

Počnimo sa grafovima!



- Zamislimo da imamo četiri čvora, i zaključili smo da su određeni parovi njih na neki način *povezani*.



- Dobili smo uobičajen **graf**; u ovom kontekstu uglavnom nazivan *monopleks mrežom*.

Još neki grafovi



- Sada primećujemo da, u nekim drugim “referentnim sistemima” (videćemo primere uskoro), ove čvorove možemo povezivati na druge načine.



Uticaj

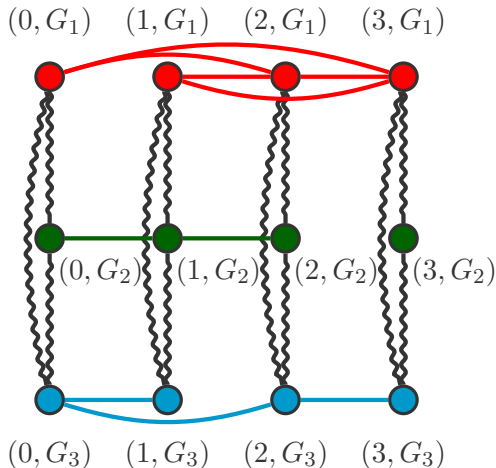


- ▶ Konačno, zaključili smo da ovi “slojevi interakcija” nisu nezavisni jedan od drugog, nego mogu međusobno interagovati na netrivialne načine (formirajući tako prethodno spomenutu “*mrežu mreža*”).
- ▶ Multipleks mreže nam daju relativno jednostavan način da predstavimo ove interakcije—dodavanjem novih *međuslojnih ivica* između čvorova i njihovih “slika” u drugim slojevima.
- ▶ Ako se vratimo na prethodni primer...

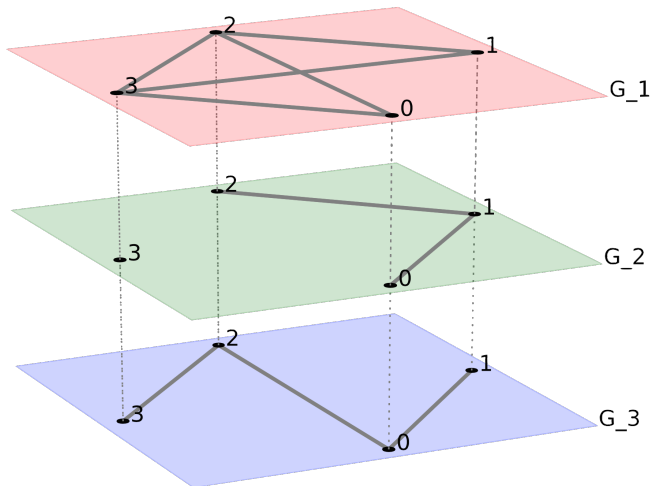
Prethodni primer



Prethodni primer



Napravili smo multipleks mrežu!



Primene na modeliranje epidemija



- ▶ Ovaj model je posebno podesan za modeliranje epidemija, pošto očekujemo da u svakom sloju posmatramo interakcije nad *istim* čvorovima (članovima populacije).
- ▶ Takođe, menjanje stanja čvora u jednom sloju potencijalno direktno utiče na njegova stanja u ostalim slojevima (što opravdava predloženi model međuslojnih ivica).
- ▶ Potencijal ovog modela je prepoznat u mnogobrojnim nedavno objavljenim radovima...
 - ▶ Granell *et al.*, 2013/2014.
 - ▶ Buono *et al.*, 2014.
 - ▶ Zhao *et al.*, 2014.
 - ▶ Azimi-Tafreshi, 2015.
 - ▶ Zuzek *et al.*, 2015.

Svrha *Viral* sistema



- ▶ Praktično svi istraživački naponi vezani za modeliranje epidemija nad multipleks mrežama do sada vrše evaluaciju na *generisanim mrežama* (Erdős-Rényi, Barabási-Albert).
- ▶ Takođe, neretko se prave pretpostavke nad slojevima koje ne moraju odgovarati ljudskoj psihologiji.
- ▶ *Viral* je napravljen sa ciljem da to promeni, i omogućiti istraživačima da:
 - ▶ Vrš *kontrolisane eksperimente* ove prirode sa stvarnim ljudima.
 - ▶ Izvuku *detaljne informacije* o stanju modela u svakom momentu simulacije, uz automatski generisane, *publication-ready* vizuelizacije;
 - ▶ Sve ovo pruži bez da bude zahtevan za uspostavljanje i detaljno konfigurisanje.

Interni model



- ▶ Razmatramo dvoslojnu multipleks mrežu, kojom modeliramo procese širenja epidemije i informisanosti, respektivno.
- ▶ Za epidemijski sloj, pretpostavljamo *SIS* (*susceptible-infected-susceptible*) proces u kome se epidemija širi *putem vazduha*; svaki čvor odašilje svoje *geografske koordinate*, na osnovu čega se određuju *verovatnoće prenosa bolesti* između svaka dva čvora.
- ▶ Za informacioni sloj, pretpostavljamo *UAU* (*unaware-aware-unaware*) proces, u kome informisanost može da se širi između čvorova koji *razmenjuju informacije*.

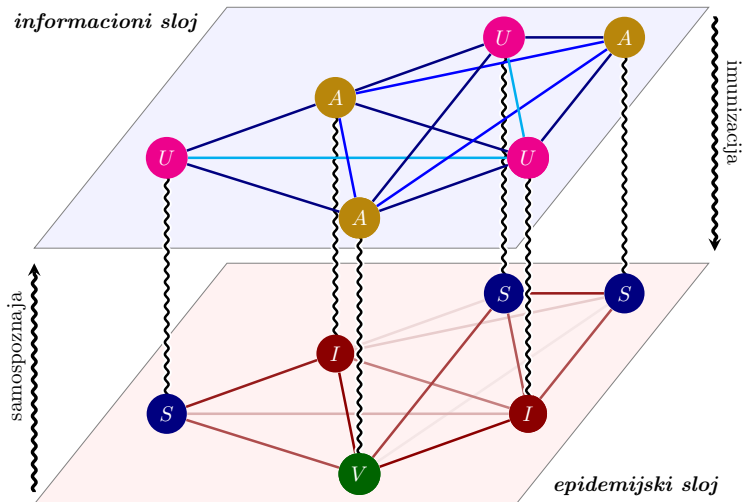
Interakcije između slojeva



- ▶ Ova dva sloja utiču jedan na drugi na *dva* kritična načina.
- ▶ Podložan (*susceptible*) čvor koji je informisan o epidemiji (*aware*) može da se *vakciniše*¹, čime potencijalno smanjuje verovatnoću zaraze.
- ▶ Podložan čvor koji postane inficiran (*infected*) će, sa određenom (fiksnom) verovatnoćom, postati svestan da se razboleo, čime može postati informisan o epidemiji (ukoliko već nije) i početi da širi informacije ostalim čvorovima.

¹**N.B.** pojmom vakcinacije ovde obuhvatamo sve moguće preventivne mere.

Izgled internog modela



Nove ideje



- ▶ Iako je *Viral* većinski implementaciono zasnovan na ustanovljenim metodama za modeliranje, tokom planiranja i razvoja aplikacije doneseno je nekoliko odluka u dizajnu koji predstavljaju potpuno *nove ideje*.
- ▶ Ove ideje su uglavnom implementirane sa svrhom da bi se:
 - ▶ Otklanjale bespotrebne apstrakcije;
 - ▶ Približavali realnom ljudskom ponašanju;
 - ▶ Smanjivala količina stanja koja mora da se pamti u modelu.

Iako su ovo pretežno rešenja za probleme praktične prirode, verujemo da bi njihovo razmatranje potencijalno bilo korisno i u potpuno teoretskom kontekstu.

Implicitan informacioni sloj



- ▶ *Problem*: direktno modeliranje informacionog sloja bi zahtevalo u najmanju ruku izgradnju neke vrste *veštačke društvene mreže* unutar naše aplikacije.
- ▶ Ovo je problematično i sa stanovišta *preciznosti modeliranja i složenosti implementacije*:
 - ▶ *Još jedna društvena mreža...* nije za očekivati da će korisnici hteti da se potpuno angažuju u njenu izgradnju; samim tim, dobijamo *nepotpune informacije* o društvenim interakcijama nad datom populacijom.
 - ▶ Uređaji koje bi ljudi koristili u eksperimentu bi morali da budu, u najmanju ruku, potpuno informisani o svim ostalim uređajima, što predstavlja rast *neophodnog stanja* koje mora da se održava u njima.
- ▶ *Rešenje*: Simulirati ovaj sloj **implicitno!**

Implicitan informacioni sloj



- ▶ Implicitno simuliranje podrazumeva da, interno, model uopšte ne održava sloj informacija, nego na početku određenom procentu populacije da neophodnu informaciju, koja onda može da se širi putem već postojećih društvenih mreža (*Facebook, Twitter...*), verbalno, itd.
- ▶ Konkretno, *informacija* u našem sistemu predstavlja **jedinstveni kod** za tekuću simulaciju. Unošenjem koda, podložni uređaji mogu postati vakcinisani.
- ▶ Ovim ne samo da smo se potpuno približili direktnom modeliranju ljudske psihologije u modernom društvu, nego i znatno umanjujemo neophodno stanje koje mora da se čuva na uređajima (mogu biti i daleko jednostavniji od telefona...).

Uloge



- ▶ *Problem:* Jedan vrlo očekivan način učešća ljudi u ovoj igri je konvergencija ka “*čoporskom ponašanju*”, u kome se informacije odmah maksimalno rašire i celokupna populacija se ubrzo vakciniše.
Ovo ne odgovara očekivanom ponašanju ljudi u stvarnom svetu!
- ▶ *Rešenje:* Direktno ohrabriti određen procenat populacije da se tokom eksperimenta ponaša *nesmotreno*, dodeljujući im jednu od dve **uloge**:
 - ▶ **Čovek** (*human*) ima cilj da ostane zdrav do kraja eksperimenta;
 - ▶ **Infektor** (*infector*) ima cilj da se do kraja eksperimenta inficira bar 50% populacije.

Uloge

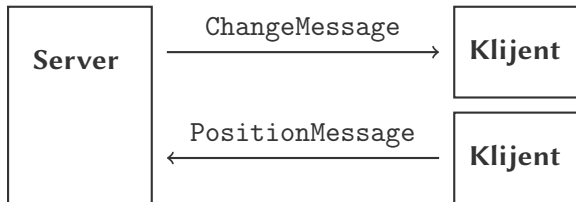


- ▶ Ovim ponovo nismo ni na koji način učinili složenijim količinu stanja koja mora da se pamti na uređajima u sistemu.
- ▶ Takođe, rezultati koje smo dobili u eksperimentima sa *sintetičkim akterima* ukazuju da dodeljivanje uloga *ne izaziva pojavu neočekivanih izlaza za karakteristične scenarije*.
- ▶ Više o tome (i o konkretnoj implementaciji *Viral* sistema) u narednim sekcijama...

Pregled komponenti aplikacije



- ▶ *Viral* je zasnovan na jednostavnom **klijent-server** distribuiranom modelu:
 - ▶ **Server** je zadužen za pokretanje *rundi*, praćenje stanja svih klijenata i komunikaciju sa klijentima.
 - ▶ **Klijenti** (uglavnom telefoni) su dostupni korisnicima i jedino imaju ulogu razmene poruka sa serverom.



Zadaci servera



- ▶ Server čuva stanja svih klijenata: **koordinate**, **fizičko stanje**, **informisanost** (uslovno rečeno), **ulogu**.
- ▶ Server potencijalno **ažurira** stanja klijenata na osnovu zadatih **parametara**.
- ▶ Server razmenjuje **poruke** sa svim klijentima.
- ▶ Server pokreće i završava runde.
- ▶ Server **beleži** stanja u kojima se multipleks mreža nalazila.
- ▶ Server proizvodi kvalitetnu **vizuelizaciju** multipleks mreže.

Ažuriranje mreže



- ▶ Server povremeno ažurira matricu $\mathbf{M}$ čija polja zavise od uzajamnih udaljenosti čvorova (matrica $\mathbf{d}$):

$$\mathbf{M}_{ij} = ke^{-\lambda d_{ij}}$$

gde su $k, \lambda > 0$ parametri mreže.

- ▶ U svakom diskretnom koraku se bira ivica $a \rightarrow b$ za **aktivaciju** u tom koraku, čime dolazi do interakcije između čvorova a i b .
- ▶ Odabir ivice se vrši po raspodeli verovatnoća. Konkretno, verovatnoća za odabir ivice $a \rightarrow b$ je:

$$\mathbf{P}_{ab} = \frac{\mathbf{M}_{ab}}{\sum_{i,j} \mathbf{M}_{ij}}$$

Interakcija između čvorova



- ▶ Kada se aktivira ivica između carrier/infected i susceptible čvora, na osnovu parametra `initialSymptomaticProbability` se određuje verovatnoća da se susceptible čvor pretvara u carrier ili infected čvor.
- ▶ Kada se aktivira ivica između carrier/infected i vaccinated čvora, na osnovu parametra `infectedIfVaccinatedProbability` se određuje verovatnoća da se vaccinated čvor pretvara u carrier ili infected čvor.

Parametrizacija servera



- ▶ U modelovanju epimedija postoje različiti nivoi *intenziteta* nekih efekata.
- ▶ Zato je server skoro u potpunosti **konfigurabilan**: parametri poput verovatnoće prenosa infekcija, početne informisanosti... zavise od vrednosti koje se mreži zadaju. Ovo uključuje i sve parametre pomenute u diskusiji o modelu.
- ▶ Ovo omogućava modeliranje raznih scenarija!

Primer rada servera



```
Socket accepted!
New node connected. {NodeId : 4, Awareness : AWARE, Physical : SUSCEPTIBLE, Latitude : 0.00000}
Accepting...
Socket accepted!
New node connected. {NodeId : 5, Awareness : UNAWARE, Physical : SUSCEPTIBLE, Latitude : 0.00000}
Accepting...
Socket accepted!
New node connected. {NodeId : 6, Awareness : UNAWARE, Physical : SUSCEPTIBLE, Latitude : 0.00000}
Accepting...
Socket accepted!
New node connected. {NodeId : 7, Awareness : UNAWARE, Physical : SUSCEPTIBLE, Latitude : 0.00000}
Accepting...
Received CodeMessage!
id = 1, Code = 5941753
Received CodeMessage!
id = 3, Code = 5941753
Received CodeMessage!
id = 4, Code = 5941753
Received PositionMessage!
5.470742543452228E-6 55.000000379030375 0.0
Received PositionMessage!
5.25355463888865E-6 55.00000063497172 0.0
Received PositionMessage!
5.5200105250012007E-6 54.00000077122505 0.0
```

Vrste poruka



HelloNewMessage	Klijent se po prvi put obraća serveru i traži identifikacioni broj.
HelloMessage	Klijent se obraća serveru nakon ponovnog uspostavljanja konekcija, kada već poseduje identifikacioni broj.
StartMessage	Server informiše klijenta da može da učestvuje u rundi, zadaje identifikacioni broj, uz početno fizičko stanje, informisanost, ulogu.
StopMessage	Server informiše klijenta da je runda gotova, kao i o tome da li je ispunio svoj zadatak u ovoj rundi, na osnovu uloge.
...	...

Vrste poruka



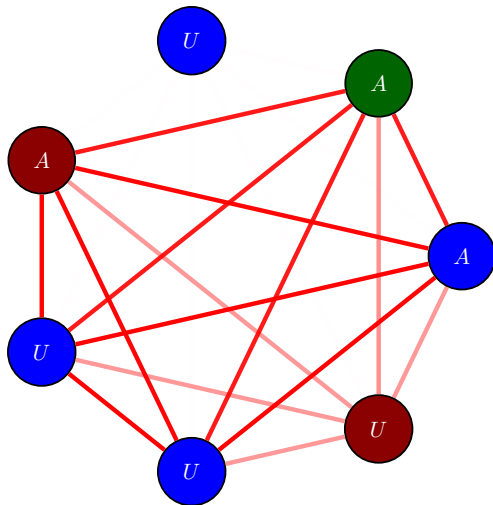
...	...
PositionMessage	Klijent šalje podatke o svom položaju (preko GPS ili mrežnih podataka).
ChangeMessage	Server informiše klijenta o promeni fizičkog stanja.
CodeMessage	Klijent šalje serveru kod za vakcinu (ne nužno tačan).

Loger i TikZer



- ▶ Istraživačima su od velike koristi faze kroz koje je mreža prolazila.
- ▶ Zbog toga **loger** kreira veliki tekstualni fajl, u kom se nalazi aktuelno stanje mreže. Za strukturu log fajlova pročitati README.md (više o tome kasnije).
- ▶ Poseban modul u okviru servera je **TikZer**, koji periodično proizvodi dijagrame koji predstavljaju trenutno stanje multipleks mreže.
- ▶ Dijagram predstavlja potpun graf, s tim što intenzitet ivica odražava blizinu čvorova (u kontekstu epidemijskog sloja).

Izlaz TikZera

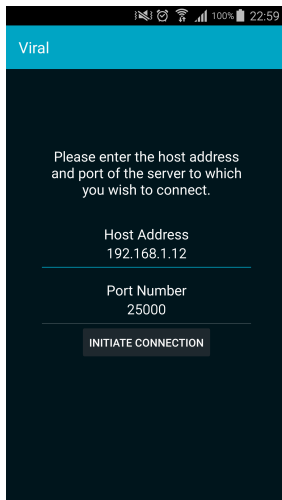


Zadaci klijenta

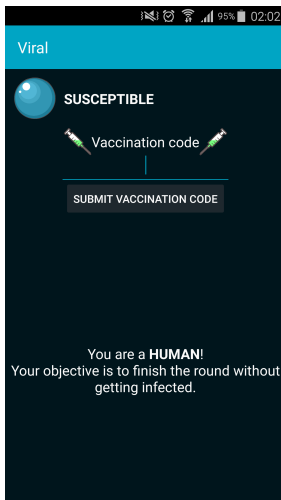


- ▶ Aplikacija na Android telefonu pruža korisniku okruženje u kom je:
 - ▶ prisutna informacija o fizičkom stanju;
 - ▶ potencijalno prisutan kod za vakcinu;
 - ▶ prisutna informacija o ulozi igrača;
 - ▶ omogućeno slanje koda za vakcinu.
- ▶ Periodično se automatski šalju poruke serveru. Ključne poruke se tiču promene položaja, koje se šalju svaki put kada sistem geolokacije na telefonu promeni ove podatke (bilo preko GPS ili mreže).
- ▶ Procesiraju se poruke koje stižu od servera.
- ▶ Aplikacija je *pozadinska*, te ne zahteva preveliku pažnju korisnika.

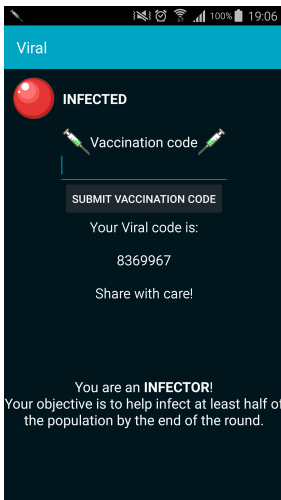
Slike aplikacije



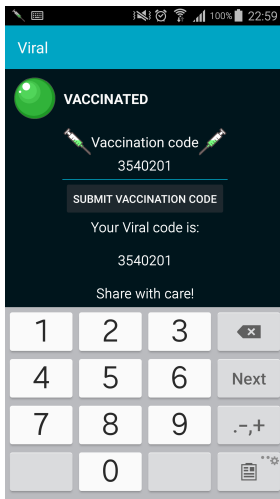
Slike aplikacije



Slike aplikacije



Slike aplikacije



Vakcine i kodovi



- ▶ Na početku runde, biraju se *informisani* korisnici koji dobijaju tačan kod za vakcinu (svako nezavisno postaje informisan sa nekom malom, određenom verovatnoćom).
- ▶ Očekuje se da korisnici informišu druge korisnike koje poznaju o ovom kodu, bilo preko društvenih mreža, bilo direktnim informisanjem.
- ▶ Ne očekujemo da će svi korisnici biti "odgovorni" i "ažurni" nakon notifikacije o početku runde!



Funkcionisanje botova



- ▶ U svrhu preliminarnih *sintetičkih* eksperimenata, kao i omogućavanja veštačkog povećanja veličine populacije, implementirali smo **botove**.
- ▶ **Botovi** su virtuelni klijenti koji šalju serveru šimulirane"podatke o položaju.
- ▶ Moguće je podešavanje nekih svojstava botova i uticati na njihovo ponašanje.
- ▶ Služe da modeluju slučajne prolaznike"ili "gužvu na trgu".
- ▶ Ne postoji način sticanja informacija, sem od servera!

Modelovanje ponašanja botova



- ▶ Svi botovi koriste model *slučajnog lutanja* po dve koordinate: geografskoj širini i geografskoj dužini.
- ▶ Ako bot nije dobio ulogu infektora, vakcinisaće se ukoliko je to moguće.
- ▶ Botovi ne stupaju ni u kakvu interakciju sa drugim botovima.

Slanje podataka



- ▶ Botovi šalju periodično i podatke o svom stanju, kao i svi klijenti.
- ▶ Na telefonima nije predvidljiva učestanost promene koordinata.
- ▶ Pretpostavka: slanje podataka je slučajan **Puasonov proces**, te je vreme T (u ms) između slanja podataka slučajna promenljiva sa eksponencijalnom raspodelom:

$$f_T(t) = \lambda e^{-\lambda t}$$

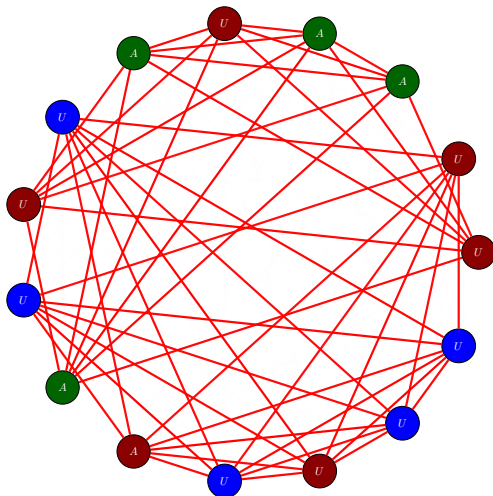
- ▶ Za svakog bota se zadaje **očekivano vreme** između slanja podataka $\mathbb{E}(T)$, odnosno $\frac{1}{\lambda}$.

Rezultati sintetičkih eksperimenata

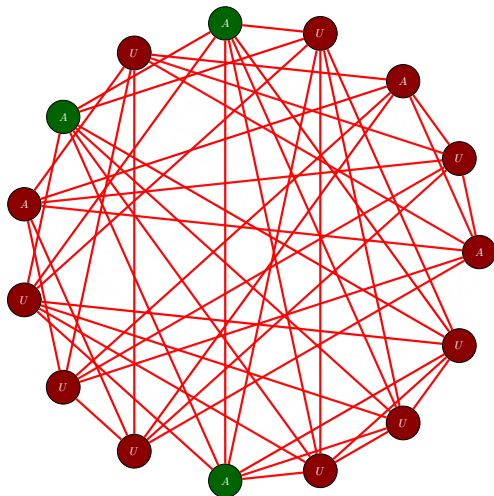


- ▶ Botovi nam omogućavaju i da napravimo preliminarne **sintetičke** eksperimente.
- ▶ Slede rezultati eksperimenata (odnosno izlaz TikZer modula), nakon variranja parametara mreže...

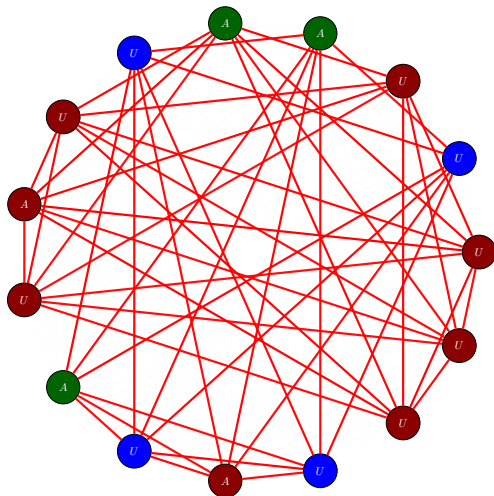
Rezultati sintetičkih eksperimenata



Rezultati sintetičkih eksperimenata



Rezultati sintetičkih eksperimenata



Uputstva za korišćenje



- ▶ Ceo izvorni kod se nalazi na GitHub-u.
- ▶ Za preuzimanje repozitorijuma izvršiti narednu instrukciju u *shell*-u:

```
$ git clone https://github.com/PetarV-/viral.git
```
- ▶ Pogledati README.md (i dalje u izradi) za detaljna uputstva za korišćenje servera, podešavanja botova, strukturu log fajlova...

Zaključci



- ▶ *Viral* je jedna potpuna *open-source* implementacija modela širenja epidemija.
- ▶ Model je zasnovan na multipleks mrežama, sa epidemijskim i informacionim slojem, pri čemu se informacioni sloj modeluje *implicitno*.
- ▶ Predviđen, pre svega, za *kontrolisane eksperimente*.
- ▶ Korišćenje i podešavanja su vrlo jednostavni.
- ▶ Nadamo se da će biti od koristi istraživačima u polju bioinformatike!