Homework 3: Simulations

Nicole Zattarin

Abstract

ciao belli

1. Single service queue

Let us consider a single server queue, we perform a discrete-time simulation under the assumption that arrivals cannot leave in the same slot in which they arrive. We consider the following situations:

(a) $P[1 \text{ arrival}] = P[2 \text{ arrivals}] = a, P[0 \text{ arrival}] = 1 - 2a$, $a \in [0, 0.5]$. and single service time for each user;

(b) $P[1 \text{ arrival}] = P[0 \text{ arrival}] = 0.5$ and geometric service time with mean probability b.

We design a program that simulate a fixed number of time slots (10000 in our case) and computes the average throughput, delay, occupancy, arrival rate, service time and returns the whole history of the queue state. Moreover, we are interested in studying the stability of the queue, to do so we compute the utilization factor ρ, which is computed as follows:

$$\rho = \frac{\text{average arrival rate}}{\text{average service time}}. \tag{1}$$

From the previous definition it is pretty intuitive to understand that the condition for stability is the following:

- $\rho > 1$ the rate of arrivals cannot be compensated with the departure rate, i.e. service time is too long to serve enough users and the state diverges;
- $\rho = 1$ the rate of arrivals is equal to the service time, thus we would expect to observe the queue periodically getting empty for a while, but after a certain time it is likely that the status diverges.
- $\rho < 1$ the service time is such that it is possibile to serve all users, thus the queue periodically empties because users arrived are quickly served.

Finally, to make our simulation more realistic, we implement a finite-size buffer system, such that if a new arrival would exceed the maximum queue size, it is dropped. This mechanism also allows to measure the overflow probability of a queueing system.

1.1. Single service time - 0/1/2 arrivals

In case (a) the average service time is constant, thus the value of ρ depends on the arrivals rate: the more is the probability of observing 1 or 2 arrivals the larger will be ρ: $\rho \propto a$. Indeed, in Figure **??** we show delay, i.e. the time a user waits in the queue until it can be served, vs ρ by varying a from 0 to 1/3. We can observe that ρ grows with the delay, while both delay and ρ increase with a .As we

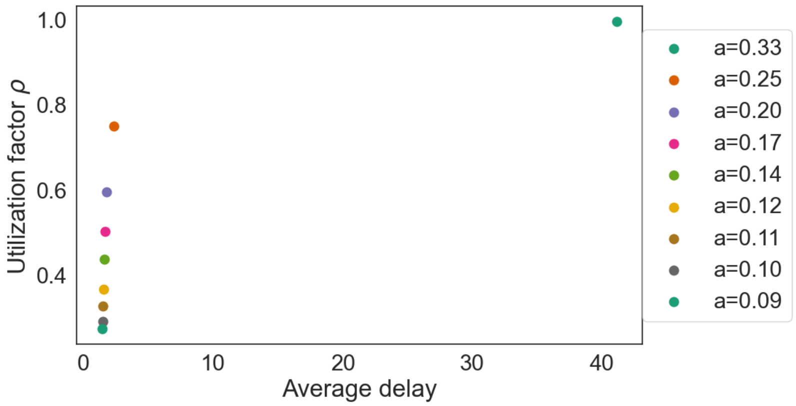


Figure 1: *Queueing system with $P[1\ arrival] = P[2\ arrivals] = a, P[0\ arrival] = 1-2a$ and unitary service time, delay vs ρ by varying a from 0 to 1/3.*

already pointed out, we would have expected to observe ρ increasing with a, since larger values of a correspond to a higher probability of observing 1 and 2 arrivals with respect to 0 arrivals. Moreover, delay increases with ρ since higher values of ρ, in the case of a fixed service time, correspond to an accumulation of users in the queue that are waiting to be served.

In Figure **??** we show the realization of queue size vs time for 10000 slots. Figure **??** shows results for $a = 1/4$, which corresponds to $\rho = 0.77$, thus to a stable queue. We can observe the expected behaviour for a stable system: the queue periodically empties, then new users are added and so on for a theoretically infinite time. On the other hand, figure **??** exhibits the same plot for $a = 1/3$, $\rho = 1$. In this case the behaviour is less stable, the queue status oscillates and the system empties once, but the general trend is increasing. Finally, Figure **??** shows the unstable simulation for $a = 1/2$, $\rho = 1.5$, in this case the probability of observing zero arrivals is null, while service time is still fixed to 1. As a consequence the number of users waiting to be served diverges.

1.2. Geometric service time - 0/1 arrivals

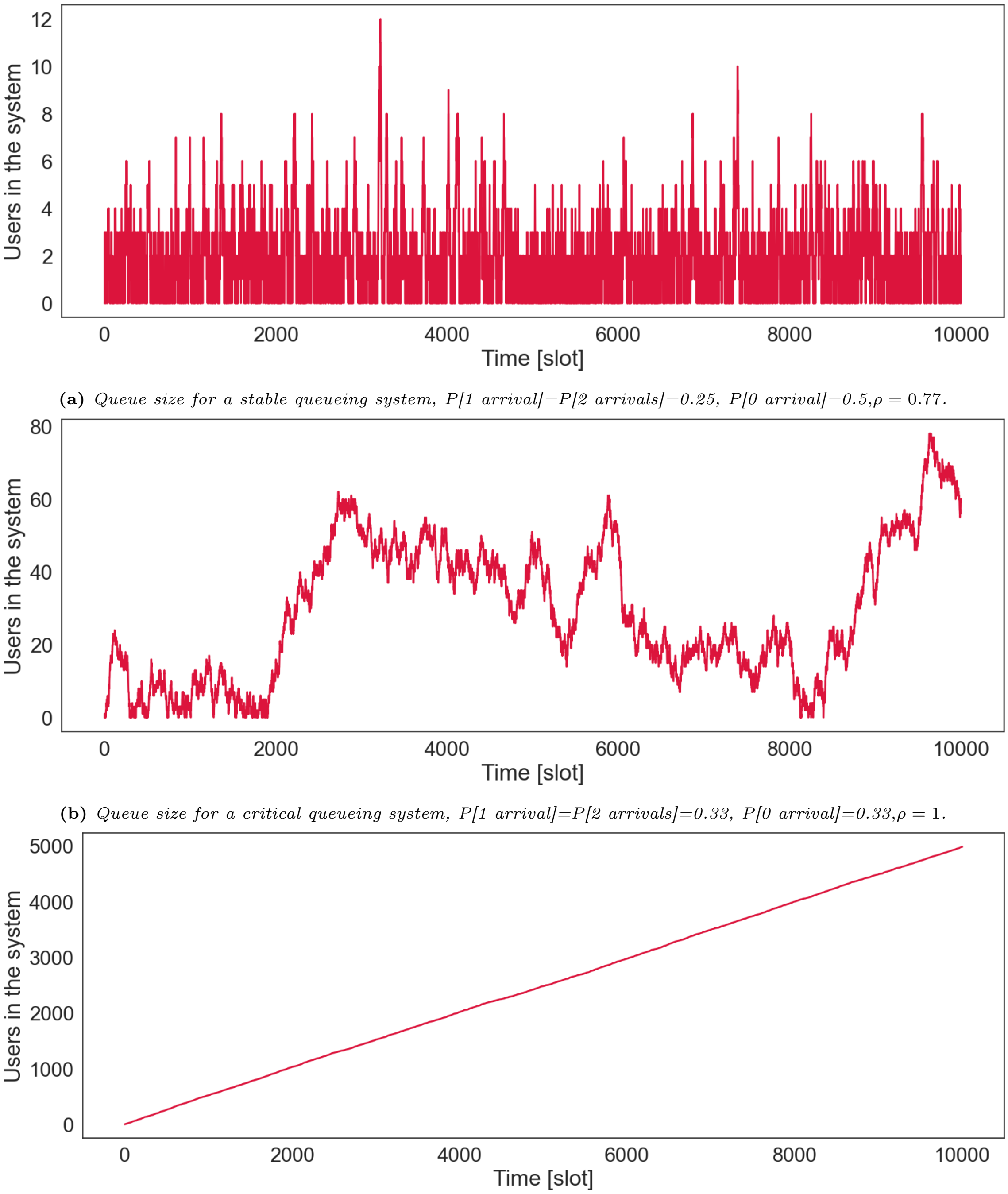
In case (b) the average service time is a geometric RV with mean value $1/b$, as a consequence the value of ρ depends on the parameter b. In this case we would expect the opposite behaviour: larger values of b correspond to larger faster service time in average and then to a faster processing of users in the queue.

In Figure **??** we show delay vs ρ by varying b from 0.5 to 1, as already introduced we observe that ρ grows with delay, while both delay and ρ decrease with b. Thus, in this case jobs are processed faster and users wait for less time in queue as b increases.

In Figure **??** we show the realization of queue size vs time for 10000 slots. Figure **??** shows results for $b = 1/3$ and $\rho = 1.50$, in Figure **??** we have $b = 1/2$ and $\rho = 1$, and n Figure **??** we have $b = 2/3$ and $\rho = 0.77$. In a completely analogous way as in case (a), but with the opposite trend, the previous cited figures denote respectively a non-stable, critical and stable queueing system.

1.3. Overflow probability

As we already pointed out, the service system is such that if the a new arrival would exceed the buffer size it is dropped. Before computing explicitly the buffer sizes necessary to guarantee a certain dropping probability, let us have a further look at Figures **??** and **??**. It is worth to highlight that



(a) *Queue size for a stable queueing system, P[1 arrival]=P[2 arrivals]=0.25, P[0 arrival]=0.5,*$\rho = 0.77$.

(b) *Queue size for a critical queueing system, P[1 arrival]=P[2 arrivals]=0.33, P[0 arrival]=0.33,*$\rho = 1$.

(c) *Queue size for an unstable queueing system, P[1 arrival]=P[2 arrivals]=0.50, P[0 arrival]=0,*$\rho = 1.5$.

Figure 2: *Realization of queue size vs time for 10000 slots for different values of 1 and 2 arrivals probabilities, fixed service time.*

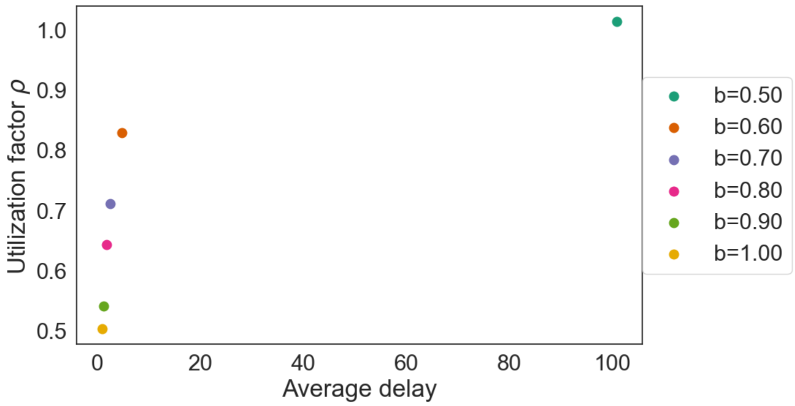


Figure 3: *Queueing system with fixed 0 and 1 user arrivals probabilities and geometric service times with probability b, delay vs ρ by varying b from 0.5 to 1*

Case	Parameter	ρ	Buffer size for 0.001% overflow
(b)	b=2/3	0.77	8
(b)	b=1/2	1	79
(a)	a=1/4	0.77	8
(a)	a=1/3	1	73

Table 1: *Values of buffer sizes which guarantee a probability* 0.001% *of overflow for different stable simulations.*

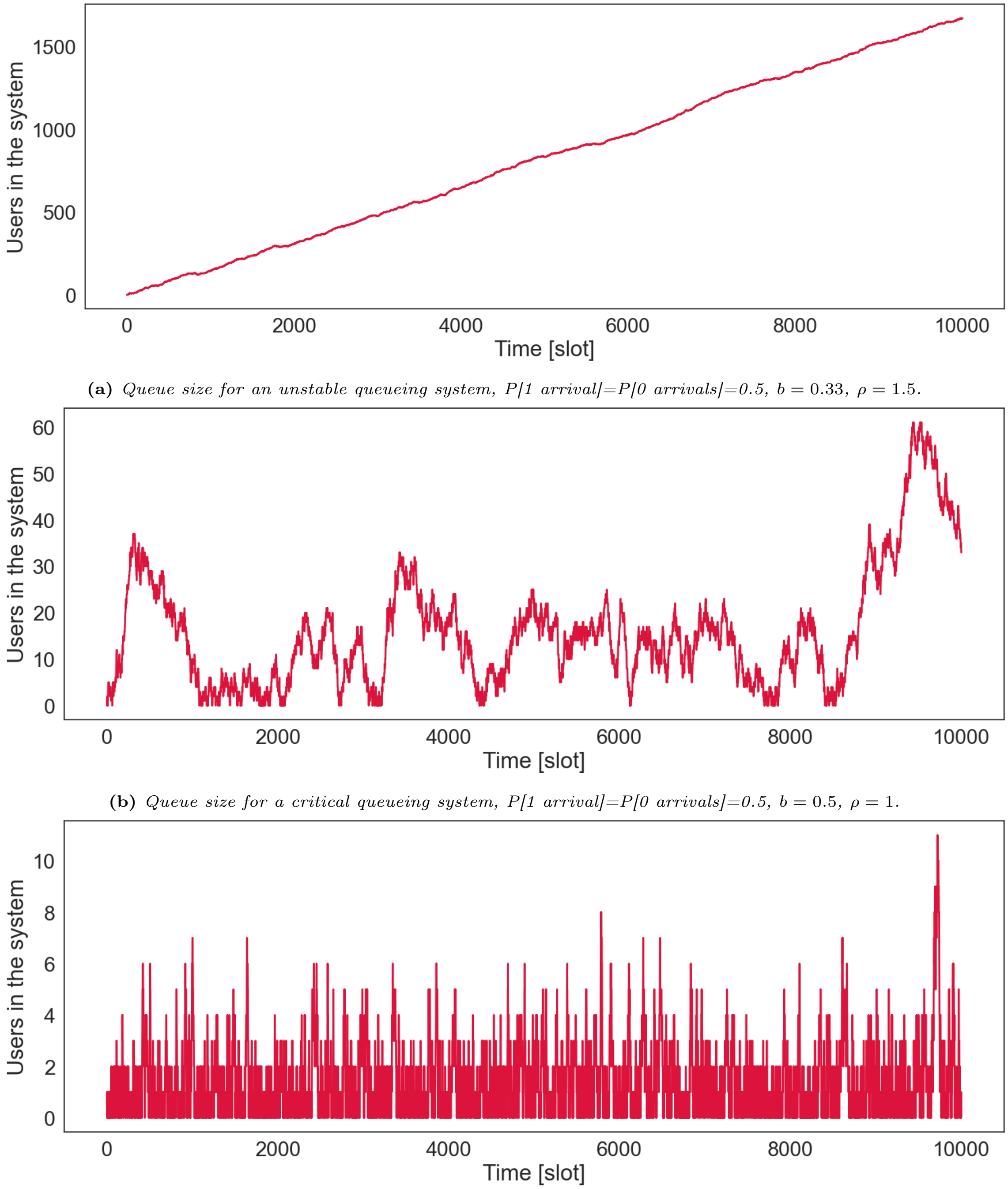
the number of users in the system is constantly lower than 12 and, apart from a few isolated points, it is in general lower than 8.

We now want to be quantitative and perform a bisection root search to find the *integer* value of buffer size which guarantees a probability 0.001% of overflow, results for the cases $\rho \leq 1$ are reported in Table **??**. The main outcome is that stable queues don't need a huge buffer capacity to provide a low dropping probability, while as the behaviour becomes more critical a larger queue is required. Finally, note also that results are similar, if not the same, for both simulations. This is not surprising, since the overflow probability depends on the nature of the system, if it is stable or not, so on the value of ρ, independently of the system employed to manage the queueing system.

2. SINR performance

A useful quantity to measure the quality of a signal is the SINR (Signal to Interference and Noise Ratio), which basically takes into account the strength of the wanted signal compared to the unwanted interference and noise. Given a communication system affected by noise and interference, we want to evaluate its performances in terms of probability to correctly deliver the information, to do so, we compute the probability that the SNR is greater than a fixed threshold b:

$$P_s = P[SNR > b], \tag{2}$$



(a) *Queue size for an unstable queueing system, P[1 arrival]=P[0 arrivals]=0.5,* $b = 0.33$*,* $\rho = 1.5$*.*

(b) *Queue size for a critical queueing system, P[1 arrival]=P[0 arrivals]=0.5,* $b = 0.5$*,* $\rho = 1$*.*

(c) *Queue size for a stable queueing system, P[1 arrival]=P[0 arrivals]=0.5,* $b = 0.67$*,* $\rho = 0.77$*.*

Figure 4: *Realization of queue size vs time for 10000 slots for fixed values of 1 and 0 arrivals probabilities, geometric service times with different probabilities.*

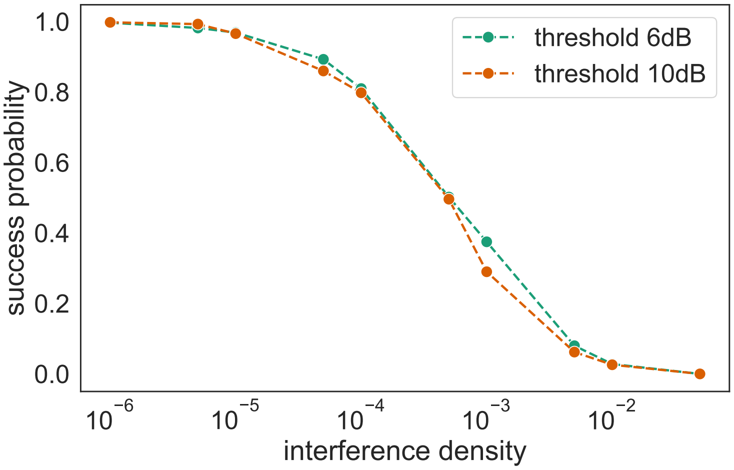


Figure 5

where we refer to b also as capture ratio. Moreover, P_s can be computed analytically as follows:

$$\begin{aligned} P_s(r_0) &= \int_{-\infty}^{+\infty} \frac{d\xi_0}{\sqrt{2\pi}\sigma} e^{-\frac{\xi_0^2}{2\sigma^2}} [I(\xi_0, r_0)]^k, \\ I(\xi_0, r_0) &= \int_{-\infty}^{+\infty} \frac{d\xi}{\sqrt{2\pi}\sigma} e^{-\frac{\xi^2}{2\sigma^2}} \int_0^1 \frac{h(r)dr}{1 + be^{\xi-\xi_0(\frac{r}{r_0})^{-\eta}}}, \end{aligned} \tag{3}$$

where ξ_0 is normally distributed with zero mean and σ^2 variance, r_0 is the position of the intended user and r is distributed as the interferers. To solve the problem, i.e. to find Equation **??**, two possible approaches are computationally feasible:

- Montecarlo simulation: randomly generate multiple times the involved variables with the required statistic and compute the frequency of the event $SNR > b$;
- Gauss Quadrature Rules to compute the integral in Equation **??**.

The system described is still generic and can be used to describe different physical systems, we study the following situation:

- Packet radio: we compute the capture probabilities , r_0 is fixed, while r is in $[0, \infty)$;
- Cellular systems: we are interested in computing the outrage probability, r_0 uniform in the cell, while r follows the distribution of interferers;
- ALOHA: we compute throughput, all users in the same area, all r in [0,1].

2.1. Packet radio

2.2. Cellular systems

2.3. ALOHA

3. GeRaF multihop performance

In this section we reproduce the results proposed in [**?**] and compare the Geographic Random Forwarding (GeRaF) model performances with GAF [**?**], see also [**?**] for further analysis. In

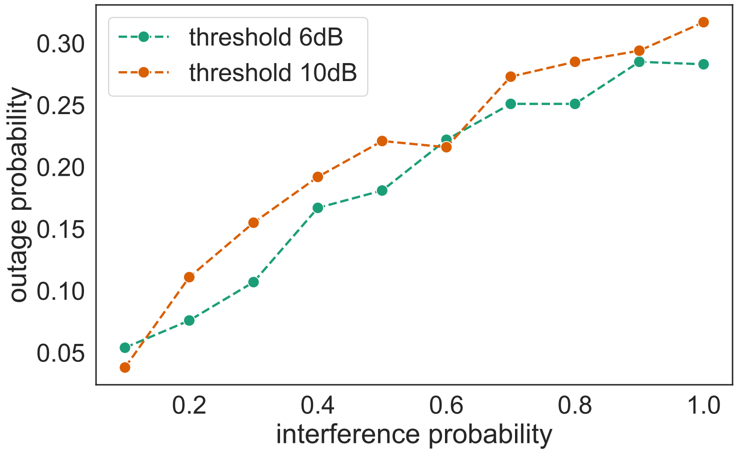


Figure 6

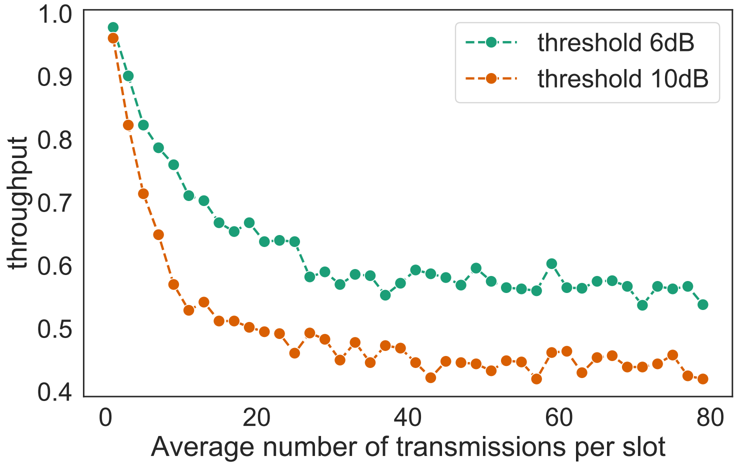
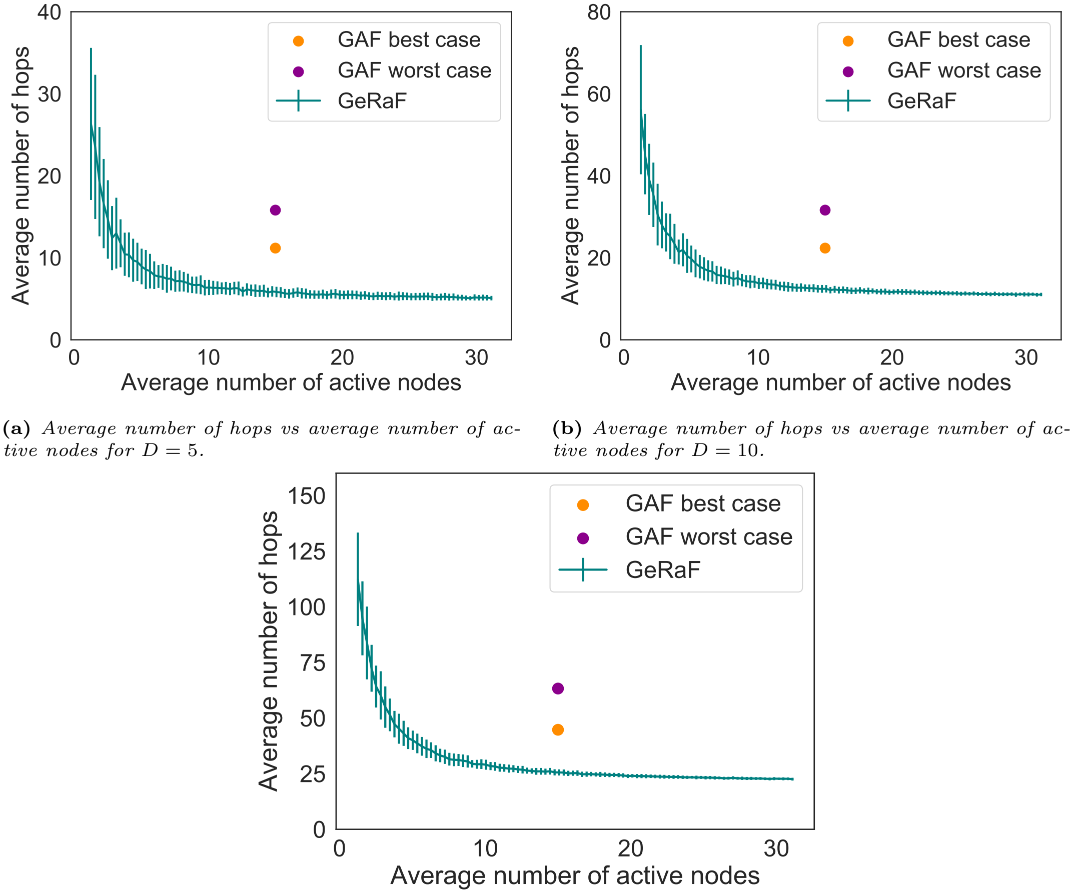


Figure 7



(a) *Average number of hops vs average number of active nodes for $D = 5$.*

(b) *Average number of hops vs average number of active nodes for $D = 10$.*

(c) *Average number of hops vs average number of active nodes for $D = 20$.*

Figure 8: *Average number of hops and respective standard deviation vs average number of active nodes for different distances between sources and destination.*

particular, we implement GeRaF algorithm for the estimation of the average number of hops needed per the forwarding. In Figure **??** we exhibit the average number of hops with the respective standard deviations vs average number of active nodes for different distances, averages are computed over a sample of 100 simulations results. In particular, we can observe that GeRaF performs better than GAF best case for a reasonable number of active nodes. Moreover, the number of hops quickly decreases up to about 10 nodes and then almost get stabilized on a value which depends on the distance, but that is still better than GAF in any case.

References

[] ZORZI M. Geographic random forwarding (geraf) for adhoc and sensor networks : Multihop performance. *IEEE Trans. Mobile Comput.*, 2(4):337–365, 2003. URL `https://cir.nii.ac.jp/crid/1571980075820873728`.

[] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, page 70–84, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581134223. doi: 10.1145/381677.381685. URL https://doi.org/10.1145/381677.381685.

[] M. Zorzi and R.R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, 2003. doi: 10.1109/TMC.2003.1255650.