Università di Pisa

Department of Information Engineering

Aeronautical Communication System

Performance Evaluation of Computer Systems and Networks project

Pietro Gronchi
Amedeo Pochiero
Giovan Battista Rolandi

A.Y. 2019-2020

Contents

1 Introduction

Aeronautical Communication System is a cummunication system between aircrafts (AC) and a control tower (CT). Connection between AC and CT is provided by ground base stations (BS), which are deployed over a grid, at a distance M from neighbors.

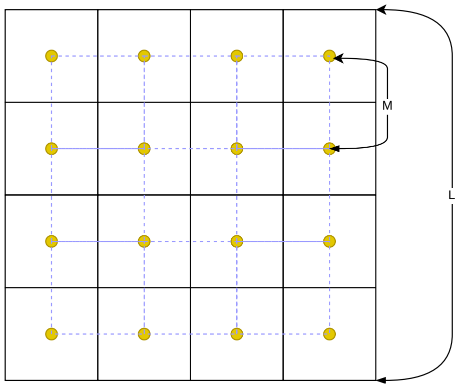


Figure 1: System topology

Deployment grid has a square topology, eg. number of rows is equal to number of columns. In Figure 1, deployment grid is drawn with dotted blue lines, while solid black lines delimit the area, called *cell*, where an AC is closer to the inner base station. L is a shortcut to indicate $M \cdot n$, where n is the number of BSs that lay on a row (or a column).
Each AC selects only one BS at a time as its serving BS, and can transmit only one packet at a time. Possibly, packets are backlogged in a queue on the AC.
ACs execute the handover procedure every t seconds, accordingly with the following algorithm:

```
if (serving BS is the closest one) then
  do nothing
else
  select closest BS as serving BS
  pause transmissions for penalty time (p)
endif
```

2 Objectives

Minimize end to end delay between AC and CT, using the maximum packet interarrival frequency, without destabilizing queues in the system.

3 Performance Indexes

- response time
- queue length

4 Scenario

- **k**: it represents the interarrival time of packets;

- **M**: it represents the distance between two consecutive BS on the same row, or column, and it is fixed at 25 km;
- **d**: it represents the distance between AC and BS;
- **s**: it represents the service time of a packet, and is given by $s = T \cdot d^2$;
- **T**: it is a tuning parameter for service time, whose value is costant for every AC in the system;
- **t**: it represents the handover period;
- **p**: it represents the handover penalty;

5 Calibration

6 Model

The ACS can be modeled with the following system from queueing theory.

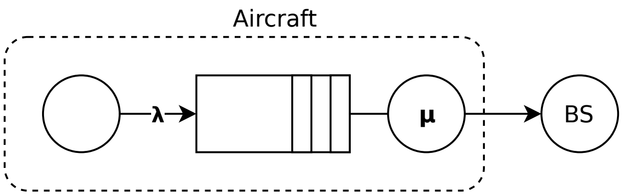


Figure 2: M/G/1 system for ACS

7 Simulator

7.1 Software

Simulator has been coded in C++ using the OMNeT++ environment and the INET framework.

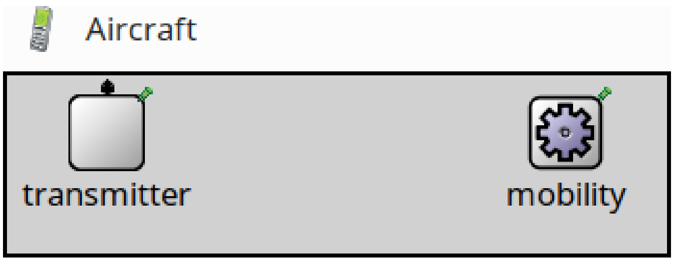


Figure 3: Module structure of Aircraft

7.2 Movement of AC

Movement of AC is given by the following algorithm, implemented using the `TurtleMobility` module from INET:

- generate a vector $\vec{s}$ such that $|\vec{s}| \in \mathcal{U}(0, M_T\sqrt{2})$ and $\angle s \in \mathcal{U}(0, 2\pi)$; upper bound of $\vec{s}$ is maximum possibile distance before AC surely wraps around;
- given that AC is in position $\vec{a}$, move to position $\vec{a} + \vec{s}$, at constant v speed, possibly wrapping around at the edges of the simulated topology;
- once reached new position, re-do from the beginning;

This algorithm is equivalent to `Random WayPoint`, but takes in account the possibility of a wrap around.

8 Verification

In order to verify the correctness of our code, we recorded the service time for a simulation and looked at a small time interval in order to spot abviously wrong situations.

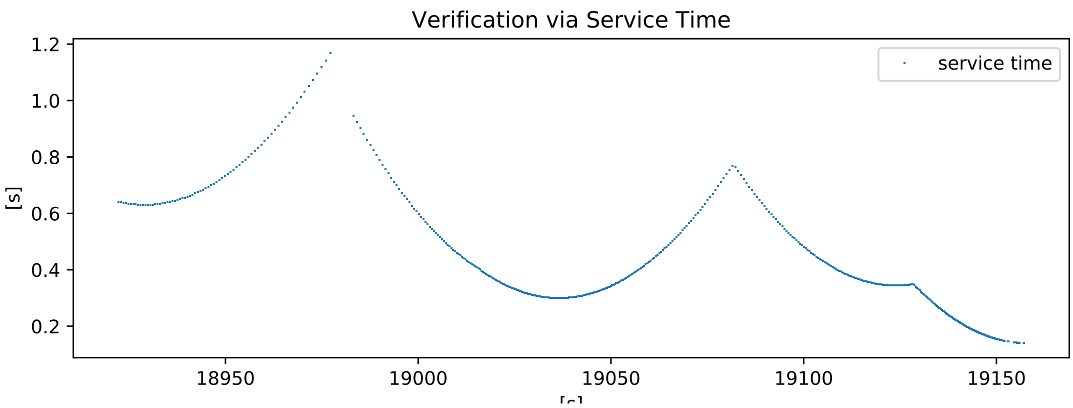


Figure 4: Service time during a test simulation

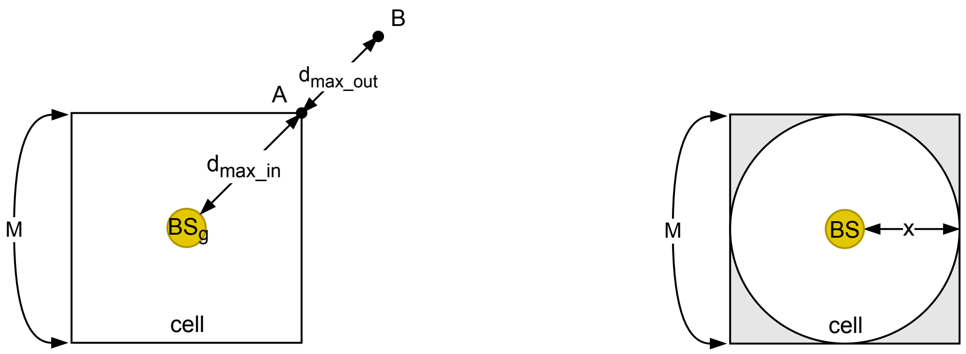
We can see that:

- service time has a paraboloid shape, which is expected since it is proportional with d^2, which d increases linearly during time, because AC travel at costant speed;
- graph discontinuity, like the one between 18950 and 19000, represent the penalty time, which can be observed after an handover; service time correctly decreases after the handover;
- graph non derivability, like two points slightly before 19100 and 19150, represent the moment when AC randomly changes direction;

At this point, we are confident that the code for our simulator is correctly modeling the real world scenario, thus we can go through a more in-depth inspection by performing some validation test.

9 Validation tests

9.1 Maximum distance considerations



(a) Computation of d_{max}

(b) Simplified distance computation

Figure 5: Computation of distance between AC and BS

Service time depends on distance between AC and BS. Distance between AC and BS, namely d, belongs to interval $[0, d_{max}[$ where d_{max} can be computed as $d_{max\_in} + d_{max\_out}$, where:

- $d_{max\_in} = \frac{M\sqrt{2}}{2}$
- $d_{max\_out} = vt$

What is the worst case? Referring to Figure 5a, let's assume that an AC is moving in direction $BS_g \rightarrow B$, and that, at a given time t_g, $\overrightarrow{AC} \in cell$ and $\overrightarrow{AC} - \overrightarrow{A} < \epsilon$ with ϵ small as you wish, eg. AC is at point A but still inside the cell where the nearest BS is BS_g; and let's assume that, at this time, AC initiates the handover procedure and discovers that the nearest BS is BS_g. Immediately after this handover procedure, performed without any penalty time, AC is out of the cell and will not initiate the handover procedure before t time: this means that, when it is in point B, it has traveled for, $d_{max\_out} = vt$, since the last handover procedure, and that the longest distance d_{max} between AC and BS is given by:

$$d_{max} = \frac{M\sqrt{2}}{2} + vt$$

This result can be used to validate our simulator.

9.2 Distances distribution simplification

In order to compute the distribution of service time, and to validate our model against a known queuing theory model, we had to make some simplifications for the computation of distance, and in particular: a single circle area around a single BS was used, as shown in Figure 5b. This way, we did not take in account for:

- handover, because its probability is difficult to compute;
- area near cell's corners (coloured area), because it introduced a non-trivial-to-manage discontinuity in our formulas;

We want to compute $P\{D = x\}$, eg. probability that, picked a random position for AC inside the cell, distance D between AC and BS equals to x. Intuition suggests that, with bigger circumferences (centered in BS) with radius x, in the real world, more points are suitable to satisfy $D = x$, thus $P\{D = x\}$ is proportional with circumference, which is proportional with radius. So, a reasonable PDF for D is:

$$f_D(x) = k \cdot 2\pi x, 0 \leq x \leq \frac{M}{2}$$

with $k = 2.03718 \times 10^{-9}$ (computed with normalization condition).
Since our goal is computing the distribution for service time, we note that:

$$F_D(x) = \int_0^x f_D(x)dx = k\pi x^2$$

and also that

$$s = T \cdot x^2 \implies x = +\sqrt{\frac{s}{T}}$$

and

$$F_S(s) = F_D\left(\sqrt{\frac{s}{T}}\right) = \frac{k\pi}{T}s$$

then

$$f_S(s) = \frac{k\pi}{T}, s \in \left[0; \frac{TM^2}{4}\right] \implies S \in \mathcal{U}\left(0, \frac{TM^2}{4}\right)$$

This result can be used to validate our simulator.

9.3 Handover convenience

Last test was on the handover convenience: it is straightforward that we want

$$s[t] \geq s[t+1]$$

for each $t, t+1$ such that and handover has been performed in between, eg. service time never increases after an handover.
This result can be used to validate our simulator.

10 Validation experiments

10.1 Maximum distance considerations

10.2 Distances distribution simplification

10.3 Handover convenience

11 Experiments