Limitations and Sidelink-based Extensions of 3GPP Cellular Access Protocols for Very Crowded Environments

Paolo Castagno[a], Vincenzo Mancuso[c], Matteo Sereno[a], Marco Ajmone Marsan[b,c]

[a] *Università degli Studi di Torino, Turin, Italy*
[b] *Politecnico di Torino, Turin, Italy*
[c] *IMDEA Networks Institute, Madrid, Spain*

Abstract

An experience common to smartphone users is the difficulty in accessing services in crowded scenarios, such as a rock concert or a football match. In these cases, to (partially) mitigate frustration, users generically claim that network congestion is occurring, and try again and again to access the network with their smartphones: the result is that user frustration and network congestion reinforce each other! This paper investigates the root causes of poor performance of cellular networks in crowded environments and shows that the commonly adopted random access procedure can prevent full utilization of wireless resources. We develop a simple yet accurate analytical model to analyze why attempting random access to wireless resources can become a problem even when access congestion avoidance is enforced, e.g., with the Access Class Barring technique. The model we propose suggests that cluster-based network access, leveraging device-to-device communications, significantly alleviates access problems. Moreover, it sheds light on scalability laws that govern network utilization and quality of experience, in terms of cell capacity, number of access channels, and cluster size.

Keywords: Crowded radio access network, 3GPP RAN, performance analysis and modelling

1. Introduction

Our common experience is that wireless access networks perform poorly in very crowded environments. When we enjoy a football match or a rock concert in an extremely crowded stadium, and we try to share our emotions with friends, we discover that placing a phone call or sending a short video, even posting a picture, is not possible, due to network congestion. When large numbers of networking experts gather at top international conferences in their field to discuss the latest research results, reading emails during the occasionally uninteresting talk is a problem, because the wireless access network is not able to sustain the very large number of email clients. These phenomena were quantitatively observed in [1], by collecting measurements over a tier-1 cellular network in the US during crowded events, and showing substantial performance degradations with respect to normal conditions.

The problem can only get worse. The Cisco Visual Networking Index forecast 2017-2022 [2] estimates that by 2022 the number of devices connected to IP networks will be about three and a half times as high as the world's population, generating an overall traffic of 4.8 ZB (equal to $4.8 \cdot 10^{21}$ B) per year. Over 70 % of this traffic will come from wireless devices, and 44% of the total IP traffic will be generated by smartphones. The total mobile data traffic in 2020 will reach 77 EB (almost $8 \cdot 10^{19}$ B) per month, with the highest volume in the Asia Pacific region (55.7% of the total), and the highest growth in the Middle

East Africa region (56%).

The 5G Infrastructure Public Private Partnership, in short 5G PPP, initiated by the European Commission, together with companies and research institutions of the field, shares those extreme visions [3]. Among the key challenges for 5G, a prominent position is given to the connection of over 7 trillion wireless devices serving over 7 billion people, and to the service of extremely crowded environments, such as a stadium, providing capacities of the order of 0.75 Tb/s over the stadium area, or an automated factory, comprising terminal densities up to 100 devices per m^2, and requiring sub-ms latency.

The work we present here looks at the performance of wireless access networks in extremely crowded environments, focusing as an example on the case of a group of 3GPP cells covering a stadium. The main contributions of our work are the following:

- We develop a simple analytical model that captures the key aspects of the behaviour of a cell and we use it to understand the main sources of poor performance.
- We validate the analytical model with detailed simulations, which prove the validity of the assumptions introduced for analytical tractability.
- We show how the model can be instrumental for a correct dimensioning of crowded cellular systems.
- We propose the adoption of device-to-device (D2D) communications [4] as a means to improve performance in extremely crowded environments, and we quantify the benefits that can be achieved with the D2D approach, showing that D2D clusters of size k are more beneficial to system performance than a costly increase of system capacity by a factor k (e.g., through the deployment of k more cells).

This manuscript extends our previous conference paper [5]. Besides refreshing the related work and the terminology used for D2D in the context of 3GPP recommendations, and adding new extensive numerical results, this manuscript provides simple yet effective approximate expressions for the notable operational points of the system, and analyzes the asymptotic behavior of the Random Access CHannel (RACH) load vs. the size of the cell population. Moreover, here we also discuss the practical validity of the proposed model, identify its limitations, and we analyze the convergence of our proposed iterative approach.

The rest of this paper is structured as follows. Section 2 discusses the stadium scenario that we consider in this work; Section 3 overviews resource allocation request procedures. Section 4 presents the analytical model. Section 5 uses the model to illustrate the system behavior. Section 6 discusses numerical results. Section 7 discusses the practical relevance of the model with respect to operating parameters that are not considered in the analysis. Section 8 addresses related work, and Section 9 concludes the paper.

2. Scenario

The reference scenario that we use in our analysis is a large stadium, with capacity roughly comprised between 50 and 100 thousand spectators. Many such structures exist around the world, including, e.g.: the Maracana in Rio de Janeiro, the San Siro Stadium in Milan, the Bernabeu in Madrid, the Stade de France in Paris, the Wembley Stadium in London, the Camp Nou in Barcelona, the Rose Bowl in Pasadena, the Azteca Stadium in Mexico City, and the the Melbourne Cricket Ground, just to name a few. These structures regularly host important sport events, and occasionally also music concerts of famous rock and pop stars, and in the latter case the structure capacity grows by up to 50 thousand attendees.

Of course, such extraordinary numbers of people (terminals) imply a wide variety of services: spectators may want to send to their friends short videos or pictures of the event, may receive all sort of messages, as well as phone calls, and at the same time terminals may be involved in content downloads.

We primarily focus on services which imply human intervention, such as the transmission of a picture with a messaging application. In this case, the human user is in the service loop, so that the basic sequence of the service operations is made of a request for the radio access network resources, possibly

automatically repeated several times, until resources are granted, then the use of the network resources, followed by a think time before the next service request.

We will see that in some cases the system bottleneck is in the request for the radio access network resources, mostly because 3GPP-compliant cellular systems use an Aloha-like contention-based scheme for this operation. It may thus happen that, while the network resources are available, request collisions do not allow their allocation. Under these circumstances, a reduction of the number of requests is mandatory to restore acceptable network performance. This can be obtained by reducing the number of users who are allowed to issue requests, or by forcing users to *coalesce* during the request phase. This is where D2D comes into play. If end user terminals are allowed to form clusters or are instructed to form clusters by the network and use *sidelinks* for intra-cluster relay—through appropriate commands issued by the cellular base station (BS), according to 3GPP standards developed starting with 3GPP release 12 and afterwards with 5G specifications [6]—only one request is issued whenever multiple terminals of the same cluster require access to the network resources, as proposed in [7] for opportunistic scenarios.

3. Accessing Resources in 3GPP Networks

In 3GPP standards like LTE, LTE-A and the upcoming 5G, end user terminals (called User Equipments – UEs in the LTE jargon) have to proceed through the Random Access CHannel (RACH) procedure to access data channels, if not already connected to the BS (called evolved NodeB – eNodeB in LTE and generalized NodeB – gNodeB in 5G). The access procedure begins with the UE sending a message on the Physical RACH (PRACH). Two types of random access procedures are defined: contention-based (implying an inherent risk of collision) and contention-free [8]. In each 3GPP-compliant cell, a fixed number (64) of orthogonal preamble signatures (PSs) are available, and the operation of the two types of RACH procedure depends on a partitioning of these PSs between those for contention-based access and those reserved for allocation to specific UEs on a contention-free basis. The contention-free RACH procedure is reserved to delay-sensitive cases, such as incoming traffic and handovers [9]. Otherwise, a contention-based random access PS is chosen at a UE to send a random access signal to the BS. A conflict occurs if more than one UE uses the same PS and time-frequency resources, resulting in undecodable messages at the BS. The contention-based procedure consists of an exchange of four messages to set up a connection among UE and BS.

Step 1: UE $\rightarrow$ BS (Random Access Preamble). A first message conveys the randomly chosen RACH PS. The UE selects one of the available PSs and transmits it in a time-frequency slot. Several UEs may choose the same PS and the BS may not be able to decode it. After the PS transmission, UE begins to monitor the downlink control channel (PDCCH) looking for an answer.

Step 2: UE $\leftarrow$ BS (Random Access Response). This message is sent by the BS on the PDCCH, and addressed with an ID identifying the time-frequency slot in which the PS was decoded. Whether multiple UEs have collided or not, if no Random Access Response (RAR) matching message has been received within the RAR window, they must repeat the RACH procedure, after a backoff delay. The duration of such backoff is randomly chosen in the range $(0, B]$ where B is the maximum number of subframes in a backoff period, and varies in $(0 - 960]$ ms.

Step 3: UE $\rightarrow$ BS (Scheduled Transmission). The UE that receives the RAR message responds with a scheduled transmission request that includes the ID of the device and a radio resource control (RRC) connection request message on the uplink shared channel (UL-SCH).

Step 4: UE $\leftarrow$ BS (Content Resolution). Contention resolution is released from the BS on the PDSCH. This identifies that no conflict on the access procedure exists. The UE can transfer data to BS.

Once a UE has successfully performed the RACH procedure, it owns an active duplex connection and is in the RRC_CONNECTED state. Keeping a connection running requires that the BS reserves physical resources devoted to this connection, even if there is no traffic available for the intended UE. Therefore the BS can handle only a limited number of connected devices. Moreover, the connected UE has to continuously verify if there is any incoming traffic, monitoring control channels, and therefore incurs high battery consumption.

As long as the communication is alive, the UE remains in the RRC_CONNECTED state, but, after an inactivity period, it begins to perform sleep cycles, from which it can return to the RRC_CONNECTED state without performing the contention-based RACH procedure.

Since the above-described access mechanism is based on a multichannel slotted Aloha, each PS representing an Aloha channel, its performance degrade beyond the threshold of 1 request/slot per PS. Hence, in dense scenarios, congestion can happen and the RACH procedure can become a system bottleneck. To alleviate congestion, state of the art solutions adopt the Access Class Barring (ACB) mechanism, which segments devices in several classes [10]. Devices within each class are managed through two parameters: the access barring probability and the barring time. With ACB, devices that are ready to attempt a random access are probabilistically *barred*, and barred devices wait for a barring time before making another barring decision, i.e., a device can be barred multiple times in a row. ACB is effective in smoothing peaks of access requests, but it does not change the RACH load under steady-state conditions. Moreover, ACB introduces a stochastic delay.

4. Analytical Model

We model the operations of n end-user terminal devices located in the same cell, under the coverage of one BS. The notation used in this paper is summarized in Table 1.

Each device generates uplink transmission requests according to the 3GPP contention-based RACH procedure briefly described in Section 3 to obtain a transmission grant from the BS. We account for the fact that the establishment of downlink flows might provide the devices with extra opportunities to obtain transmission grants, skipping contention through the contention-free RACH procedure.

Table 1: Notation and Cell Parameters used in Section 6

Quantity	Notation	Value
Number of devices (or clusters)	n	
BS capacity	C	150–1500 [Mb/s]
Network max accepted requests	M	200
Number of Random Access preambles	N	54
Slot time	τ	0.01 [s]
Backoff time RACH	B_0	av. 0.15 [s]
Backoff time Network	B_1	av. 1 [s]
ACB access probability	p_a	0.05–0.95
ACB barring time	B_a	av. 4-512 [s]
Transmitted data volume	F_S	av. 1.5 [MB]
Transmission time	S	
Think time	T_{TH}	av. 30 [s]
Device uplink speed limit	R	
Probability to skip RACH procedure	p_J	≤ 0.5
Access delay	A_T	
Thinking subsystem throughput	λ	
Network subsystem throughput	ξ	
Random Access subsystem input	γ	
Arrival rate at Network subsystem	σ	
Collision probability	p_C	
Rejection Probability	p_B	

In the following, we derive a simple model for access requests and service operation in the cell, and show how to compute network utilization, access delay, and in general how to assess the behavior of the system as a function of the number of devices in the cell, for a given BS configuration (in terms of capacity, number of RACH channels, RACH slot duration, backoffs experienced upon failed RACH procedures, etc.).

4.1. Closed representation of the system

The BS has uplink capacity C, in bits per second, and can share its capacity among at most M devices at a time (i.e., there can be up to M devices in state RRC_CONNECTED). The number of RACH channels (i.e., orthogonal preamble signatures - PS) available for Random Access is N and the interval between two consecutive Random Access Opportunities (RAOs) is τ seconds. If during τ a single device selects a given RACH channel, then the RACH procedure is successful, otherwise the RACH channel is

either unused or a collision happens with multiple devices attempting to use the same PS.

A RACH collision results in a random backoff B_0, after which a RACH retry follows. In case of successful RACH procedure, the device is granted transmission only if there are less than M devices under service at the BS, otherwise the device goes through a random backoff B_1 followed by another RACH procedure. The model also considers ACB with uniform access probability p_a for all classes, and barring time with average duration $E[B_a]$ seconds.

For what concerns the traffic generated by end-user terminals, we consider human-operated wireless devices, and assume that each device produces a new transmission request, with random data volume F_S, only after its previous request has been served. More specifically, upon service completion, we assume that the devices enters a "think time" period with random duration T_{TH} before generating the next request. Unless otherwise specified, the average service time $E[S]$ only depends on C, M and the average value $E[F_S]$, i.e., we assume that the serving speed is fixed and equal to C/M, so that $E[S] = \frac{M \cdot E[F_S]}{C}$. We do so because our analysis concentrates on finding the range of values for the user population n that allows to use most, if not all, transmission resources while incurring low delay in the attempts to access the network. Therefore, we particularly focus on the network behavior under (quasi-) saturation conditions. However, we will also show how to extend the model to capture the behavior of a non-saturated system, and in particular we will show how to keep the model tractable while improving the approximation on $E[S]$ by account for: (a) equal sharing of the BS capacity among the actual number of devices under service in the system, and (b) service speeds limited by a device uplink speed R.

The resulting system model is depicted in Fig. 1. The model comprises 6 main components: *i)* Think, representing the end-user think time between the end of a service and the generation of a new access request; this is modeled with an infinite server queue with random (not necessarily exponential) service time with average $E[T_{TH}]$ seconds; *ii)* Random Access, representing the RACH contention-based procedure; this is modeled as a set of N parallel slotted Aloha channels, receiving each $\frac{1}{N}$ of the total load offered to the RACH; the slot duration for any of the N slotted Aloha channels is τ; *iii)* Barring Time, which models ACB operation as an infinite server with average service time $E[B_a]$ seconds, affecting a portion $1 - p_a$ of the flow directed to the Random Access; *iv)* Network, representing the BS resources, modeled as an M/G/M/0 queue with average service time $E[S]$ seconds. The Network queue is fed by the output of the Random Access subsystem and by the requests that skip the Random Access because of transmission opportunities generated by downlink traffic requests; these are modeled by means of the "jump probability" p_J, which is the probability to skip the contention-based RACH procedure, and access directly the BS resources. *v)* Network Backoff, and *vi)* RACH Backoff, representing the two backoffs, which are modeled by means of infinite server queues with random service times (the assumption of an exponential pdf for service times is not necessary, but may be an adequate choice to represent the behavior of real systems), with averages $E[B_0]$ and $E[B_1]$ seconds, respectively.

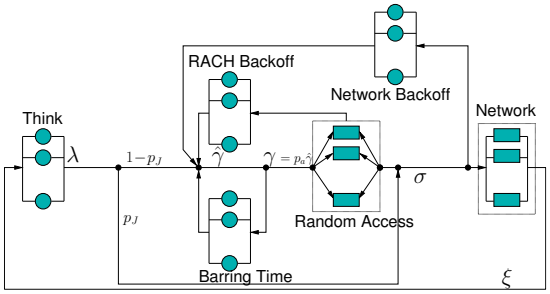

Figure 1: Closed queueing network model of a cell.

Fig. 1 also shows that the system is closed, i.e., the population is finite, with the number of customers fixed to n. We denote by λ the output of the Think subsystem, and by ξ the output of the Network subsystem. Because of the closed structure of the system, $\lambda = \xi$. We indicate with γ the total arrival rate at the N RACH channels in the Random Access subsystem, and we assume that RACH requests follow N parallel and i.i.d. Poisson arrival processes with intensity $\frac{\gamma}{N}$ arrivals per second. Although devices decide to send RACH requests asynchronously, such requests

are cumulated over τ seconds and physically sent at the same time over the same frequency band. Thus, the successful output of each of the N RACH channels is that of a slotted Aloha system with $\frac{\gamma\tau}{N}$ arrivals per slot, which is given by $\frac{\gamma\tau}{N}e^{-\frac{\gamma\tau}{N}}$ successes per slot, as known from the standard analysis of multichannel slotted Aloha [11]. The maximum throughput per slot of such multichannel slotted Aloha system is $\frac{N}{e}$, which is achieved for $\gamma\tau = N$.

With the above, the arrival rate at the network service is $\sigma = \gamma e^{-\frac{\gamma\tau}{N}} + p_J\lambda$, the arrival rate at the RACH backoff B_0 is $\gamma\left(1 - e^{-\frac{\gamma\tau}{N}}\right)$, and the one at the Network backoff B_1 is $p_B\sigma$, where p_B is the blocking probability, given by the Erlang-B formula with M servers and load $\rho = E[S]\sigma$. The load accepted and served by the network service is $\xi = (1 - p_B)\sigma$. For analytical tractability, we introduce the simplifying assumption that all arrival processes are homogeneous and independent Poisson processes. The impact of such Poisson assumptions will be assessed with simulations.

In the described system, quantities λ, σ, and ξ (and therefore also ρ and p_B) are functions of γ. It is possible to write a recursive equation in γ by considering that γ is $\hat{\gamma}$ minus what enters the Barring Time block. $\hat{\gamma}$ results from the sum of four arrival rates: $\lambda(1 - p_J)$ from the Think subsystem, the output of backoffs B_0 and B_1, plus the recycle caused by ACB:

$$\hat{\gamma}=\lambda(1-p_J)+\gamma\left(1-e^{-\frac{\gamma\tau}{N}}\right)+p_B\left(\gamma e^{-\frac{\gamma\tau}{N}}+p_J\lambda\right)+(1-p_a)\hat{\gamma},$$

which, combined with $\gamma = p_a\hat{\gamma}$, yields a recursive expression for γ, which does not depend on ACB operation at all:

$$\gamma = \lambda(1-p_J) + \gamma\left(1-e^{-\frac{\gamma\tau}{N}}\right) + p_B\left(\gamma e^{-\frac{\gamma\tau}{N}}+p_J\lambda\right). \tag{1}$$

The recursive expression Equation (1) has two unknowns: γ and λ (note that p_B can be written as function of ξ, and $\xi = \lambda$). Unfortunately, this expression is not enough to identify the operating point of the system, because it contains no dependence on the population size n. However, to introduce n in the loop, and remove λ, we can apply Little's law to different blocks in the modeled system, as presented in the following.

Solving system equations requires iteration, whose proof of convergence is provided in Section 4.6.

4.2. Dependence on the population size n

From the model described in the previous subsection, we can easily derive the expressions for the network utilization, the number of devices under service and in any of the system blocks depicted in Fig. 1, the time of a complete cycle between two transmissions, and the delay to access the service. All these quantities can be expressed as function of γ, and γ can be expressed as function of the population size n.

Utilization and distribution of devices. The network utilization ξ is equal to $\sigma(1 - p_B) = \left(\gamma e^{-\frac{\gamma\tau}{N}} + p_J\lambda\right)(1 - p_B)$. Therefore, since $\xi = \lambda$, it is immediate to obtain the following expressions for ξ, λ, σ and ρ:

$$\xi = \lambda = \frac{\gamma e^{-\frac{\gamma\tau}{N}}(1 - p_B)}{1 - p_J(1 - p_B)}; \tag{2}$$

$$\sigma = \frac{\xi}{1 - p_B} = \frac{\gamma e^{-\frac{\gamma\tau}{N}}}{1 - p_J(1 - p_B)}; \tag{3}$$

$$\rho = E[S]\sigma = \frac{E[S]\gamma e^{-\frac{\gamma\tau}{N}}}{1 - p_J(1 - p_B)}. \tag{4}$$

Note that, since ρ in Equation (4) only depends on γ and p_B, we have that p_B actually depends only on γ. Thus, all the quantities representing arrival rates in the system model are functions of γ only, for fixed values of the other system parameters.

The average number of devices under service, that cannot exceed M, is computed by applying Little's law at the Network, i.e., $n_S = \xi E[S] \leq M$, which also implies that utilization cannot exceed $M/E[S]$. Similarly, the average number of devices in Think is proportional to the average number of devices under service, i.e., $n_{TH} = \xi E[T_{TH}] = n_S\frac{E[T_{TH}]}{E[S]}$.

The rest of the devices $n - n_S - n_{TH}$ are attempting access, either waiting for the next RACH opportunity (including after a barring event) or in one of the backoff queues, so applying again Little's law we obtain:

$$n-n_S-n_{TH}=\gamma\left(\frac{\tau}{2}+\frac{1-p_a}{p_a}E[B_a]\right)+$$
$$+\gamma\left(1-e^{-\frac{\gamma\tau}{N}}\right)E\left[B_0\right]+\frac{p_B\gamma e^{-\frac{\gamma\tau}{N}}}{1-p_J\left(1-p_B\right)}E\left[B_1\right],$$

where the average delay incurred in a RACH attempt is computed as half of the slot duration because of the Poisson arrival assumption. The total number of devices in the network can therefore be expressed as a function of γ:

$$n=\gamma\left(\frac{\tau}{2}+\frac{1-p_a}{p_a}E[B_a]\right)+\gamma\left(1-e^{-\frac{\gamma\tau}{N}}\right)E[B_0]+E[B_1]$$
$$\cdot\underbrace{\frac{p_B\,\gamma e^{-\frac{\gamma\tau}{N}}}{1-p_J\left(1-p_B\right)}}_{\frac{p_B}{1-p_B}\xi}+\left(E\left[S\right]+E\left[T_{TH}\right]\right)\underbrace{\frac{\gamma e^{-\frac{\gamma\tau}{N}}\left(1-p_B\right)}{1-p_J\left(1-p_B\right)}}_{\xi}. \tag{5}$$

This is a monotonic relation between n and γ, which can be inverted (although not in closed form) to express γ as a function of n. However, we have seen that all quantities of interest in the system are functions of γ, so that we can conclude that they are eventually functions of n only, i.e., of the device population's size.

Cycle duration. The average time for a complete cycle in the system (e.g., the cycle between two consecutive service completions) is denoted by $E\left[T_{cycle}\right]$ and can be easily computed from the model of Fig. 1, by considering that: *i)* the probability to collide on a slotted Aloha representing the RACH channel with Poisson arrivals of intensity $\frac{\gamma\tau}{N}$ arrivals per slot is $p_C=1-e^{-\frac{\gamma\tau}{N}}$, and *ii)* collisions are assumed to be independent. Hence, we can write that:

$$E[T_{cycle}]=\frac{p_B}{1-p_B}E[B_1]+E[S]+E[T_{TH}]+\left(\frac{1}{1-p_B}-p_J\right)$$
$$\cdot\left[e^{\frac{\gamma\tau}{N}}\left(\frac{1-p_a}{p_a}E[B_a]+\frac{\tau}{2}+E\left[B_0\right]\right)-E[B_0]\right]. \tag{6}$$

The term in brackets in Equation (6) is the average time spent in the loop formed by the RACH and the RACH backoff blocks, which has to be counted $\frac{1}{1-p_B}$ times on average (i.e., the average number of Bernoulli trials before a success, including the success that occurs when a device finds the Network available), except for the case in which a request skips the RACH, which occurs with probability p_J. The quantity $\frac{1-p_a}{p_a}E[B_a]+\frac{\tau}{2}+E\left[B_0\right]$ is the time to complete one of such RACH loops—which includes, on average, $\frac{1-p_a}{p_a}$ passages through the ACB backoff—and there are, on average, $\frac{p_C}{1-p_C}=e^{\frac{\gamma\tau}{N}}-1$ collisions before a successful RACH attempt (in which case the RACH backoff does not occur). The network backoff is traversed only after a failed network access (i.e., $\frac{p_B}{1-p_B}$ consecutive times, on average), whilst the Network and Think subsystems are traversed only once per cycle. $E\left[T_{cycle}\right]$ depends on γ since we have shown that p_B also depends on γ. So, using Equation (5) we conclude that $E\left[T_{cycle}\right]$ can be written as function of n.

Access delay. The access delay, indicated as $E\left[A_T\right]$, is the time spent in a cycle, excluding the think time and the service, and is therefore easily obtained from Equation (6):

$$E\left[A_T\right]=E\left[T_{cycle}\right]-E[S]-E\left[T_{TH}\right]. \tag{7}$$

An alternative expression for $E\left[A_T\right]$ is obtained by applying Little's law to the part of the system that excludes network service and think time:

$$E\left[A_T\right]=\frac{n-n_S-n_{TH}}{\lambda}. \tag{8}$$

Since $\lambda=\xi$, Equation (8) reveals that the access delay is (practically) linear with the population size if ξ is (roughly) constant in a range of n, so that also n_S and n_{TH} are constant. As we will show later, such range exists if the Network saturates before the Random Access. That range is very relevant, because any point in it leads to maximal utilization.

4.3. QoE indexes

We use two indexes to express the quality of experience (QoE) for the end-user. The first index η_S compares the service time with the time spent waiting before service starts, and it decreases with the access delay:

$$\eta_S:=\frac{E[S]}{E[S]+E\left[A_T\right]}. \tag{9}$$

The second index is η_A, which is inversely proportional to the service time and fades exponentially with the access delay. Service time and access delay used in η_A are normalized to their values obtained with the smallest population n that causes the presence of M devices under service (denoted by n'):

$$\eta_A := \frac{E[S]|_{n=n'}}{E[S]} e^{-\frac{E[A_T]}{E[A_T]|_{n=n'}}}. \quad (10)$$

Differently from η_S, index η_A is very sensitive to relative increases of delay rather than to absolute increases.

4.4. Analysis with D2D support

When D2D is used to alleviate RACH contention problems, terminal clusters come into play, each of them behaving as a single device. Thus, we can use the same formulas as above, with n, n_S, n_{TH} denoting the number of clusters in the system, under service and in think time, respectively. Similarly, all arrivals and services refer to clusters. The main effect of clusters is the reduced load to the Random Access. The impact is non-linear because γ does not scale linearly with n.

Cluster formation. Clusters form either spontaneously, when a device announces its willingness to wait for other users in its close proximity to join in a random access attempt, or under the control of the BS, when RACH collision probability becomes problematic. In the latter case, the BS can oversee the formation of clusters of users located close to one another.

Service time with clusters. If k is the average cluster size, i.e., the average number of devices in a cluster, the average service time per network access request becomes k times higher than for the case without clusters, so as to be able to serve k transmissions with one request.

Device think time with clusters. Clusters are ephemeral entities, i.e., they do not persist after entering the Think subsystem; they are rather dismantled, and devices are re-shuffled to form new clusters built at random from the set of devices in the Think subsystem. Hence, there is no per-cluster think time but rather a per-device think time that depends on cluster formation rules. In the case of clustered RACH access, the average think time experienced by each device increases as well. Indeed, within each cluster, the effective think time of the device that initiates the cluster corresponds to its own think time, plus the time needed for the other members to join, because a cluster-cumulative RACH message is sent only after all cluster members have completed the thinking procedure. If a cluster of size k is formed, the device that initiates the cluster suffers the highest additional think time, due to the wait for $k-1$ devices to join; the second member of the cluster must wait for $k-2$ devices to join, and so on. Only the last (k-th) device to join does not suffer any increase in the think time. Considering the first device, and assuming a very high density of devices that are in the condition of joining the cluster (the scenario is very crowded), and assuming exponentially distributed think times, forming a cluster of a few units is very quick. The worst case average additional think time is computed for the first device in a cluster of k formed from a population of $m \gg k$ devices that can join the cluster as $\sum_{i=1}^{k-1} \frac{E[T_{TH}]}{m-i} \simeq E[T_{TH}] \left(\frac{k-1}{m}\right)$. A similar conclusion can be reached in the case of generally distributed think times, by considering an asymptotic normal approximation (thanks to the large number of potential cluster members) and the $(k-1)$-st order statistics distribution. In practice, the per-device think time increase due to clustering is negligible in crowded environments, and so we neglect it in the model and assume that for each cluster that enters the Think subsystem, another cluster leaves after a think time equal to the one of a single device.

4.5. Impact of resource sharing under non-saturated conditions

If we consider that the BS resources can be shared by active connections, it is obvious that underloaded systems offer higher rates to the active devices.[1]

[1] We remark that a cluster is seen as a single device and that the order in which cluster members' transmissions are scheduled is not important for our model, and is anyway out of the scope of the article. A cluster here is associated with

Therefore, the analysis proposed so far is valid in the region in which the Network subsystem is fully loaded (which is the focus of this paper), while it contains an approximation elsewhere. To fix this approximation, let us consider a Network subsystem that shares equally its resources among the connected devices, up to a rate R that can be interpreted as the maximum rate achievable by a device or as the maximum rate specified in the user's service level agreement. In such case, the service time conditional to j active connections becomes:

$$E\left[S|j\right] = E\left[F_S\right] \max\left\{\frac{1}{R}, \frac{1}{C/j}\right\}.$$

Such a dependence on the number of active connections requires the use of a processor sharing (PS) model serving up to a maximum number of users, offering each user up to a maximum service rate. While the resulting analytical model remains tractable using the class of Whittle Networks [12], the added complexity can be seen to have a marginal impact, especially in the operating regions that are the focus of this paper. Thus, we only examine this case through simulation, and we use an approximation to handle the case in which the Network subsystem is not fully loaded, letting:

$$E\left[S\right] = E\left[F_S\right] \max\left\{\frac{1}{R}, \frac{1}{C/n_S}\right\}. \tag{11}$$

That is, we associate with every user a capacity equal to the total capacity divided by the average number of users in the Network subsystem.

Note that $E\left[S\right]$ is equal to $\frac{E[F_S]}{R}$ when the number of devices under service is not enough to saturate the Network subsystem. The adaptation of $E[S]$ to the number of devices under service introduces a further element of dependence on n, and a non-linearity. Although its impact on system performance is quite limited in most of the cases, the use of (11) is helpful when the Network subsystem approaches saturation, so we will use it in the rest of the paper.

a message that consists in the concatenation of its members' messages.

4.6. Convergence

With constant $E[S]$. For a given user population n, we find iteratively γ, p_B and λ, and then compute all other quantities. More specifically, we use two nested iterations. First, we note that we can evaluate the value of n that corresponds to an input γ by using Equation (5) and compare it with the target. As we will show later in Equation (19), γ and n are asymptotically proportional for large values of the population, with γ being upper-bounded by n times a constant. Thus, for fixed n, the search range for γ is $[0, \omega n]$, with $\omega = \frac{\tau}{2} + \frac{1-p_a}{p_a} E[B_a] + E[B_0]$. However, the evaluation of Equation (5) requires to know the value of p_B, which is *not* a monotonic function of γ. For fixed γ, the value of p_B can be computed with a nested iteration. Specifically, we use the Erlang formula for p_B and Equation (2) for λ, thus yielding the following system of equations to find p_B (and λ at the same time, as a byproduct):

$$\begin{cases} p_B &= \frac{\left[E[S]\left(\gamma e^{-\frac{\gamma\tau}{N}} + p_J\lambda\right)\right]^M / M!}{\sum_{j=0}^{M}\left[E[S]\left(\gamma e^{-\frac{\gamma\tau}{N}} + p_J\lambda\right)\right]^j / j!}; \\ \lambda &= \frac{\gamma e^{-\frac{\gamma\tau}{N}}(1-p_B)}{1-p_J(1-p_B)}. \end{cases} \tag{12}$$

On the one hand, increases of λ correspond to increases of p_B in the first expression in Equation (12) because, for fixed γ, λ is proportional to the load of the Network subsystem $\rho = E[S]\left(\gamma e^{-\frac{\gamma\tau}{N}} + p_J\lambda\right)$. On the other hand, increases of p_B correspond to monotonic decreases of λ computed with the second expression in Equation (12). With the above property in mind, the value of λ can be found by using a dichotomic search in the interval $\left[0, \gamma e^{-\frac{\gamma\tau}{N}}/(1-p_J)\right]$, which is the range of λ when $p_B \in [0,1]$ and γ is fixed. Once λ is computed, also p_B is determined, which means that the value of p_B is univocally determined by the value imposed for γ in the current iteration. At that point, we can go back to adjust γ by comparing the target value of n with Equation (5) computed with the values of γ and p_B of the current iteration. This comparison tells us if we have to increase or decrease γ for the next iteration. Specifically, if the difference is above a specified tolerance, γ will be adjusted proportionally to the difference between the target value of n and its value resulting at

the current iteration. Moreover, if an increase (or decrease) of γ is required, then the lower (resp. upper) limit for the search range for future iterations can be set to the current value of γ. This guarantees convergence, since the search interval keeps shrinking at any iteration.

With $E[S]$ depending on n_S. In this case, to the iteration on γ with nested iteration on p_B, we need to add another nested iteration on $E[S]$, Specifically, $E[S]$ has to satisfy the following recursive equation obtained from Equation (11) with a number of devices under service computed with Little's result as $n_S = \lambda E[S]$:

$$E[S] = E\left[F_S\right] \max\left\{\frac{1}{R}, \frac{\lambda E[S]}{C}\right\}. \qquad (13)$$

Therefore, for fixed γ, we can start with the worst case for $E[S]$, which is $E[F_S]/(C/M)$ under fully saturated network conditions, and then proceed to solve Equation (12) as in the case for fixed $E[S]$, so as to update p_B and λ, and hence iterate on $E[S]$ using Equation (13). After the first iteration, since $\lambda E[S] = n_s \leq M$, the value of $E[S]$ is either the same as before—and hence no further iteration is required—or less, which means that p_B decreases, and thereby λ increases in the next iteration, thus leading to a further decrease of $E[S]$ to compensate λ, because the average population n_S cannot increase with respect to previous iteration (because the service time was taken as the worst case). The same applies at every iteration, and since $E[S]$ is bounded, the iterations converge.

Remarks. The convergence of the described iterations is typically quite fast. For instance, consider that, for the numerical results shown later, we have set the convergence threshold to one part per million with respect to increments of variables over which we iterate, and all results contained in a figure were obtained in a few tens of seconds using a laptop equipped with a dual core 3 GHz Intel i7 processor and 16 GB of RAM). However, there are points at which the variation of n with γ can be very fast, which leads to possible numerical errors in the computation of $E[S]$, when it is not fixed, and therefore of p_B and λ. This happens unless the resolution used for γ is made very fine (i.e., using 12 digits for expressing its decimal part).

5. System Behavior

Here we study the bottlenecks of the system, point out some notable points in the performance curves, and analyze how performance is affected by the number of devices present in the cell and by the introduction of D2D-based clusters.

5.1. Bottlenecks

The model depicted in Fig. 1 has two potential bottlenecks: the Random Access and the Network subsystems. The former filters network access attempts, and asymptotically prevents any network request as γ grows with the population n. The Network subsystem has finite capacity, and therefore cannot serve more than M simultaneous requests.

Fig. 2 shows a typical case in which the maximum throughput of the Random Access is below the capacity of the Network subsystem, and thus is the only bottleneck, for all population sizes. In this case, the Network subsystem throughput ξ and the input σ of the Network subsystem are equal, since the blocking probability p_B is negligible. From Equation (7), the access delay becomes a linear affine function of $e^{\frac{\gamma\tau}{N}}$, and therefore grows with e^n. However, as shown in Fig. 2, a system in which the Random Access saturates before the Network subsystem does not suffer high delay. The range of device populations that roughly maximizes network utilization is quite narrow, and corresponds to a rather small interval around the peak efficiency of a multichannel slotted Aloha system, i.e., to values of n close to the one that yields $\gamma\tau = N$ (about 400 in the figure). This is the context that was previously analysed in the literature for the case of machine to machine (M2M) communications [13], with a system model similar to ours, and studied by means of a Markov chain. Here we focus on the more complex two-bottleneck case, in which the Network subsystem saturates before the Random Access, which is typical for the stadium scenario.

Fig. 3 shows an example of the model behaviour when both the Random Access and the Network

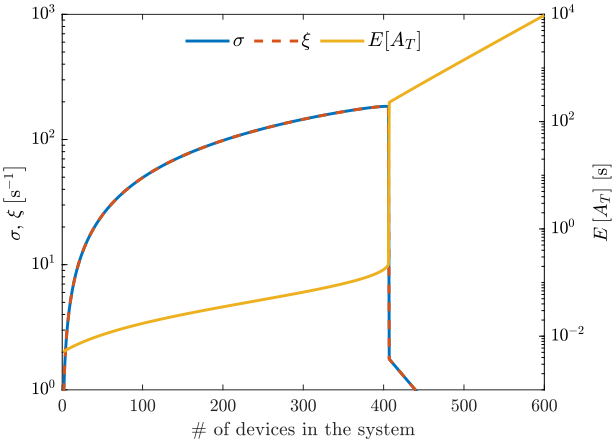

Figure 2: Random Access-limited model behaviour. Left scale for σ and ξ, right scale for access delay.

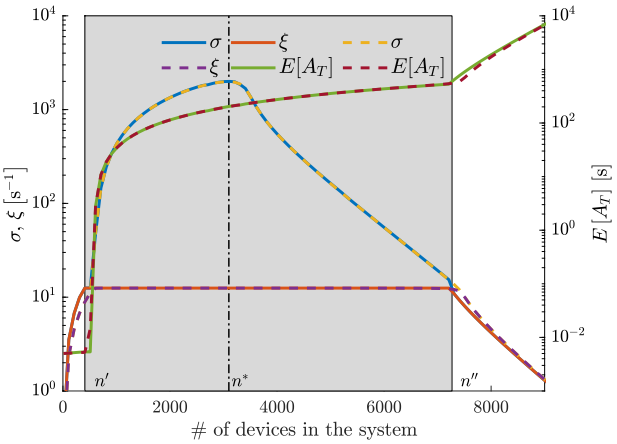

Figure 3: Model behavior with Network subsystem saturation. Left scale for σ and ξ, right scale for access delay. Dashed lines are computed with the worst case value for $E[S]$, while for generating the solid lines we have used the approximation (11).

subsystem can become the system bottleneck. Indeed, the Network subsystem is a bottleneck for lower values of population size, until the Random Access reaches success probabilities too low to starve the Network subsystem. In the figure we can identify three operational regions. In the first region (low number of devices and low load: roughly below 550 devices for the specific example), p_B and p_C are close to zero, $\sigma \simeq \xi$, and the delay is practically negligible. In the second region (shaded in the figure, roughly from 550 to 7500 users), the throughput of the Network subsystem is constant, while σ follows the familiar bell-shaped curve of slotted Aloha, and the delay grows linearly with the user population, as visible from Equation (8) (note the logarithmic vertical scale on the right). In the third region, p_B is negligible again, so that $\sigma \simeq \xi$ like in the first region, but the delay now grows exponentially with γ, and therefore with n. Out of such three regions, only the second one is desirable for system operation, since the Network subsystem resources are not wasted, and delay scales linearly with the number of devices in the cell.

5.2. Notable operational points

Random Access saturates first.[2] In this case, $p_B \simeq 0$, so that $\xi \simeq \frac{\gamma e^{-\frac{\gamma\tau}{N}}}{1-p_J}$, the average number of devices in service is $n_S = \xi E[S] < M$, and the average service time $E[S]$ must be equal to $\frac{E[F_S]}{R}$ since the Network is not saturated and every device under service obtains its maximum allowable rate R. In this scenario, the Network throughput is maximal when the output of the Random Access is maximal. This occurs for a number n^* of users that results in $\gamma = \frac{N}{\tau}$. From Equation (5) we obtain the approximation (linear in N):

$$n^* \simeq \frac{N}{2} + \frac{N}{\tau}\left[e^{-1}\frac{\frac{E[F_S]}{R} + E[T_{TH}]}{1-p_J} + (1-e^{-1})\,E[B_0] + \frac{1-p_a}{p_a}E[B_a]\right].$$

With Network saturation.[3] In this case, to characterize the behavior of the system in the three operational regions shown in Fig. 3, in addition to

[2] In this example, we use very few preamble signatures and a very high transmission capacity, so that the Random Access will saturate well before the Network. The configuration used is as follow: $N = 5$, $C = 600$ Mbps, $p_a = 1$ and $p_J = 0$.

[3] This example uses a relatively high number of preamble signatures and limited capacity, so that the Network will saturate before the Random Access. Specifically, in this case we use $N = 54$, $C = 150$ Mbps, $p_a = 1$ and $p_J = 0$.

n^* we characterize n' and n'', i.e., the values of n that correspond to the first and the second knee of the curve representing ξ vs. n. The figure reports throughputs and access delay computed with the model using the constant worst case approximation for the service time $E[S]$ (dashed lines) or the iterative approximation (11) (solid lines). The differences between the two cases are minimal. However, the first knee of the throughput curves is slightly different, and using the constant approximation leads to overestimate the point at which the Network subsystem saturates. A similar effect, although less pronounced, can be noticed for the value of n'',

Note that $n' \leq n^* \leq n''$, and the throughput of the Network subsystem is constant and equal to $\frac{C}{E[F_S]}$ for all values in the interval $[n', n'']$. Therefore, Equation (2) reduces to:

$$\gamma e^{-\frac{\gamma\tau}{N}} = \frac{C}{E\left[F_S\right]} \frac{1 - p_J\left(1 - p_B\right)}{1 - p_B}, \quad \forall \gamma \mid n \in [n', n''].$$

At the extremes of the considered interval $[n', n'']$, the Network subsystem has exactly enough resources to satisfy the demand, so that we can consider $p_B \simeq 0$:

$$\gamma e^{-\frac{\gamma\tau}{N}} \simeq \frac{C}{E\left[F_S\right]}\left(1 - p_J\right), \quad \gamma \mid n \in \{n', n''\}. \quad (14)$$

Considering that the L.H.S. of Equation (14) is a non-negative continuous function of γ that starts from 0, grows until it reaches the value $\frac{N}{e\tau}$ at $\gamma = \frac{N}{\tau}$ and then decreases asymptotically to 0, expression Equation (14) admits two (possibly coinciding) real solutions only if $\frac{C}{E[F_S]}\left(1 - p_J\right) \leq \frac{N}{e\tau}$. So, a range of values of n such that the throughput of the Network subsystem is constant and maximal exists if and only if

$$N \geq \frac{e\tau C}{E\left[F_S\right]}\left(1 - p_J\right). \quad (15)$$

The distance between the zeros of γ in Equation (14) decreases logarithmically with C increasing (and with p_J decreasing). Since γ is monotonic with respect to n, this means that the interval $[n', n'']$ becomes smaller with larger capacities C (and with smaller probabilities p_J), and $n' = n'' = n^*$ when Equation (15) holds as equality. If Equation (15) does not hold, the Network subsystem cannot saturate, and we fall back to the Random Access-limited scenario of Fig. 2.

The above condition also tells that the number of RACH channels needed to allow network saturation scales linearly with the capacity of the network and with $(1 - p_J)$.

The notable points described above and the asymptotic behavior of γ vs. n can be approximated by means of the following closed form expressions that can be readily derived, as shown next.

Approximated values for the notable points. The value of n' can be approximated in closed form if the network saturation throughput is much less than the maximum Random Access throughput. In this case, there is neither network blocking nor collisions (i.e., $p_B \simeq 0$ and $p_C \simeq 0$) at n' and $\xi = \lambda = \sigma = \frac{C}{E[F_S]}$ while $\sigma \simeq \gamma + p_J\lambda$. Considering that in this case the service time becomes $E[S] = \frac{M \cdot E[F_S]}{C}$, the result is that Equation (5) reduces to:

$$n' \simeq M + \frac{C}{E\left[F_S\right]}\left[E\left[T_{TH}\right] + \left(1 - p_J\right)\left(\frac{\tau}{2} + \frac{1 - p_a}{p_a}E[B_a]\right)\right]; \quad (16)$$

Therefore, n' scales linearly with network capacity, slot duration, and probability of skipping the Random Access.

For what concerns the value of n^*, we can again use Equation (5) with $\gamma\tau = N$, $\xi = \frac{C}{E[F_S]}$, $\rho = E[S]\left(\frac{N}{e\tau} + p_J\frac{C}{E[F_S]}\right)$. Considering that $\rho \gg M$ if the network saturates well before the Random Access, then $p_B \simeq 1 - \frac{M}{\rho}$ and $n_S \simeq M$. In conclusion, the following approximation holds:

$$n^* \simeq M + E\left[T_{TH}\right]\frac{C}{E\left[F_S\right]} + \frac{N}{2} + \frac{N}{\tau}\left(1 - e^{-1}\right)E\left[B_0\right] + \frac{N}{\tau}\frac{1 - p_a}{p_a}E[B_a] + \left[\frac{N}{e\tau} - \frac{C}{E\left[F_S\right]}\left(1 - p_J\right)\right]E\left[B_1\right]. \quad (17)$$

Therefore, the value of n^* scales linearly not only with C, but also with M, N, p_J and τ^{-1}.

The value of n'' has to be computed numerically. From Equation (5) computed for $p_B \simeq 0$ and with

the product $\gamma e^{-\frac{\gamma\tau}{N}}$ given by Equation (14), it results that:

$$n'' \simeq n' + \left[\gamma'' - \frac{C}{E\left[F_S\right]}\left(1 - p_J\right)\right] \cdot \left(\frac{\tau}{2} + \frac{1-p_a}{p_a} E[B_a] + E\left[B_0\right]\right), \qquad (18)$$

where γ'' is the largest root of Equation (14). One can notice that while n' increases linearly with C and $1 - p_J$, the distance $n'' - n'$ decreases logarithmically with the same parameters, due to Equation (14). Therefore, n'' decreases with C and $1 - p_J$ for small values of such parameters, where the log decrease is superlinear, and then increases when the logarithm becomes sublinear. Moreover, n'' grows with N, because N increases the L.H.S. of Equation (14), and therefore it has the effect of spacing apart the zeros of that equation, while n' is not affected by N, as noticed before with Equation (16).

The accuracy of the approximations for n' n^* and n'' can be appreciated by looking at Fig. 3. It can be clearly seen that the approximate values shown on the x axis match very well the model behavior observed in the three curves.

The behavior of n' and n'' vs. C is shown in Fig. 4 for the same example discussed in Fig. 3 (in there the cell capacity was fixed to $C = 150$ Mb/s, while now we consider more values). The behavior of n' is clearly linear, whereas n'' initially decreases and afterwards grows. Overall, n'' exhibits a quadratic behavior. In all cases, the distance $n'' - n'$ diminishes with C.

Asymptotic behaviour of γ vs. n. For $n > n''$, the Network throughput starts to vanish exponentially, and the result is that the device population progressively moves to the Random Access block. Asymptotically, there are no devices either in service or in think time, so that all devices loop between the ACB block, the Random Access subsystem and its backoff:

$$\lim_{n\to\infty} \frac{n}{\gamma} = \frac{\tau}{2} + \frac{1-p_a}{p_a} E[B_a] + E[B_0]. \qquad (19)$$

Fig. 5 shows the behavior of γ vs. n for the example of Fig. 3. In the figure, γ grows very slowly

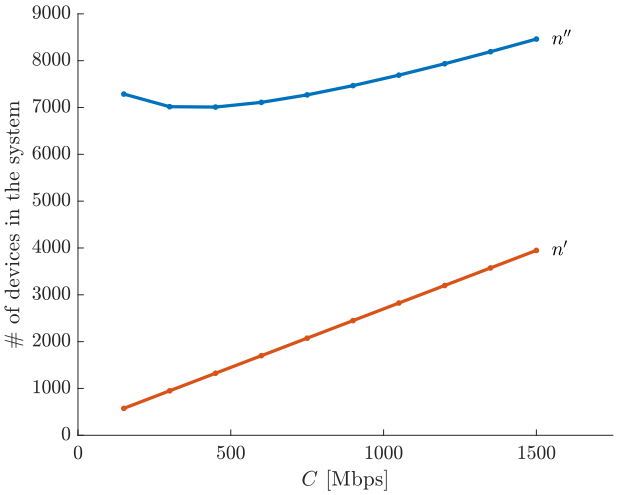

Figure 4: Behavior of n' and n'' versus the cell capacity (computed without ACB).

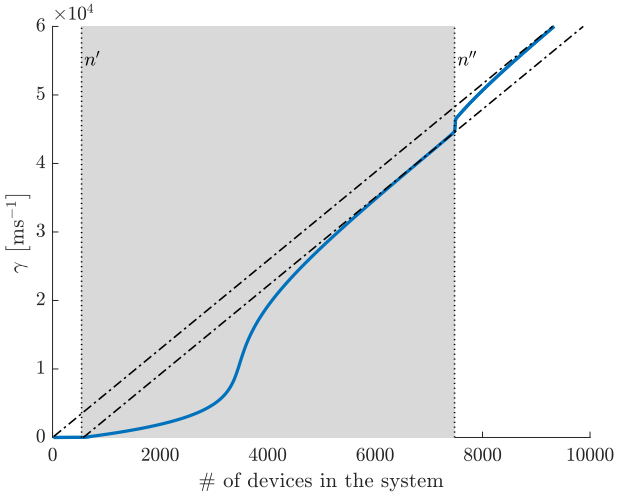

Figure 5: Monotonic relation between γ and n (computed without ACB).

for $n < n'$ (access requests do not collide and there is practically no blocking at the Network). Between n' and n'', the value of γ grows faster and faster, especially in the zone in which the RACH throughput has a negative slope (this is due to high collision probability and high Network blocking). However, as soon as p_B decreases again, due to excessive colli-

sions, the curve of γ vs. n changes towards a linear relation (right before n'', where $p_B \simeq 0$). After n'' all devices move to the Random Access and backoff B_0 subsystems (no device will be under service, asymptotically), and eventually the relation between γ and n approximates Equation (19).

With clusters. As explained in Section 4.4, clustering k devices results in transferring $kE[F_S]$ bits per network access, hence the cluster service time $E[S]$ becomes k times longer. So, n' decreases with increasing cluster size. However, the number of devices within clusters becomes kn'. Denoting by $E[S|1]$ the service time without clusters, we have:

$$kn' \simeq \frac{C}{E[F_S]} \Bigg[kE[S|1] + E[T_{TH}] + (1-p_J)\left(\frac{\tau}{2} + \frac{1-p_a}{p_a} E[B_a]\right)\Bigg], \qquad (20)$$

which includes $(k-1)\frac{CE[S|1]}{E[F_S]}$ more devices w.r.t. the case without clusters. Similarly, we can observe that kn^* grows by M plus a number of devices proportional to N for each increase of 1 in the cluster size k.

The interval $n''-n'$ increases with the cluster size, because a factor k appears in the denominator of the R.H.S. of Equation (14) when clusters are used. Therefore, the increase of the size of the network saturation region, in terms of devices, becomes $k(n''-n')$, which is more than a k-fold increase.

We can conclude that the beneficial impact of clustering is larger than the one obtained by increasing cell capacity, which is linear, and it comes at a much lower deployment cost.

5.3. Delay

The access delay $E[A_T]$ is negligible when the Random Access saturates first, and for $n < n'$ when the Network subsystem also saturates, unless ACB introduces high delay by using low values for p_a and/or high values for $E[B_a]$. When the Network subsystem is saturated, we know from Equation (8) that $E[A_T]$ is proportional to n with coefficient $\frac{1}{\lambda} = \frac{E[F_S]}{C}$. For $n > n''$, the delay explodes exponentially. Therefore, the desirable range of population sizes goes from n' to $n' + \Delta n$, where Δn is such that the delay $\frac{\Delta n E[F_S]}{C}$ is bearable by the applications running at the devices in the network.

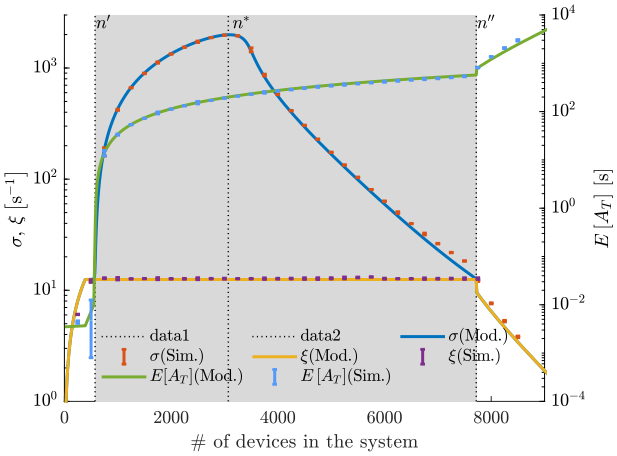

Figure 6: Model validation for a cell with 150 Mb/s capacity and $p_a = 1.0$. Left scale for σ and ξ, right scale for access delay.

So, in practice, the study of n' and its approximation are key to tune system parameters properly during network design.

5.4. Validation through packet-level simulation

In order to validate the simplifying assumptions that we had to introduce for the analytical tractability of the model, we developed an event-based packet-level simulator that reproduces the behaviour of the closed model in Fig. 1. However, in the simulator we used uniformly distributed (rather than exponential) file sizes; the arrival processes are not Poisson, and the output of the Random Access subsystem is an impulsive process in which all successful RACH attempts are brought at the Network subsystem ingress at the same time. In addition, the simulator accurately implements the processor sharing of the Network processor capacity among active connections, i.e., it equally allocates resources to active jobs (i.e., packets under service) based on the actual number of jobs, with a maximum per-job rate equal to R.

Fig. 6 reports an example of the simulated results for σ and ξ, together with the analytical results.

Specifically, we report numerical results for a cell with $C = 150$ Mb/s, $R = 10$ Mb/s, $N = 54$, $M = 200$, $p_J = 0.3$ and $\tau = 0.01$ s, which are typical values for LTE BSs. Moreover, we used $E[T_{TH}] = 30$ s, $E[B_0] = 0.15$ s, $E[B_1] = 1$ s, $E[F_S] = 1.5$ MB, $p_a = 1$ and $E[B_a] = 4.0$ s to account for typical upload of pictures and small videos during crowded events by using applications like WhatsApp, with automatic file upload retry. To ease the reader, such configuration will be used through the rest of the paper and, only when required, we will point out the parameters that deviate from the configuration presented above. We run each experiment a sufficient number of times to obtain small 95% confidence intervals. The figure clearly shows that the model is extremely accurate. The access delay computed with the model for values smaller than n' and greater than n'' underestimates the one computed via simulation by a small quantity. This is due to the fact that the service time is overestimated in those regions, so that the number of users in the RACH is (slightly) underestimated. Similarly, we can notice that the RACH throughput and the throughput of the Network subsystem are a bit underestimated for values close to and greater than n'', where the service rate is actually better than what used in the model.

We tested a wide range of values for all relevant parameters, and found very similar model accuracy in all cases. The extremely good match between model predictions and simulation results is a clear indication of the model validity beyond the simplifying assumptions introduced for tractability. The figure also shows that the approximations used to compute the notable operational points in close form are quite accurate. Some minor error can only be noticed in n', whose approximated value is slightly higher than the one obtained with the full model (which approximate the processor sharing operation in the computation of $E[S]$) and confirmed by the simulations (which uses a pure processor sharing). In fact, since the approximated expression for n' is derived by using the worst case service time $E[S]$, which in turn leads to overestimate the size of the population that saturates the Network subsystem.

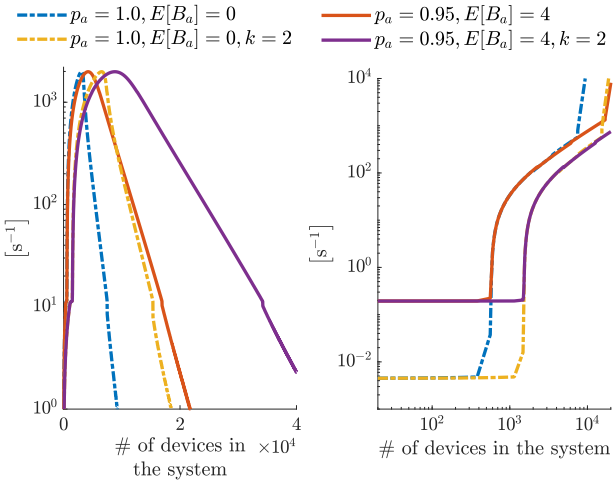

Figure 7: Throughput (left) and access delay (right): impact of ACB with [$p_a = 0.95, E[B_a]=4$] (solid lines) and clustering (where k is specified) in the stadium scenario.

6. Stadium: Numerical Results

We consider a stadium covered by a set of LTE cells. The system parameters are as reported in Table 1 (right-most column). Fig. 7 illustrates the impact of ACB and clustering in the specified scenario. We only report the results obtained with the ACB configuration that causes less delay, and one example of clustering ($k=2$). The figure shows that either ACB or clustering make it possible to significantly increase the number of users in the system. In particular, clustering groups of as few as 2 users is very effective in increasing n'. ACB suffers large delays, so as to make it quite undesirable even for limited user population sizes. However, Fig. 7 also shows that ACB and clustering *in combination* achieve low delay and guarantee access to very large user populations.

Fig. 8 reports the values for the two QoE indexes η_S and η_A we defined in Section 4.3. Both indexes capture the user satisfaction, combining the service time and the access delay. In the first case we just compute the ratio between the service time and the sum access delay plus service time. In the second case we define a more elaborate parameter. It is inversely proportional to the service time, normalized to the service time when M users are under service.

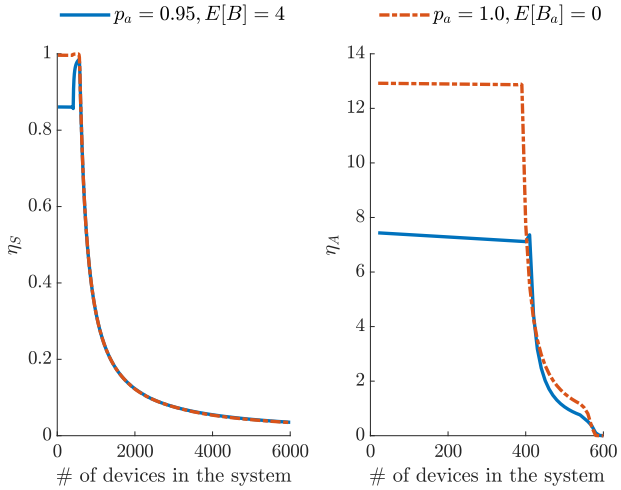

Figure 8: End-user QoE indicators η_S and η_A. Quality degrades with ACB [$p_a = 0.95, E[B_a] = 4$] (solid lines) w.r.t. scenario without ACB (dashed lines), because of the additional delay it causes.

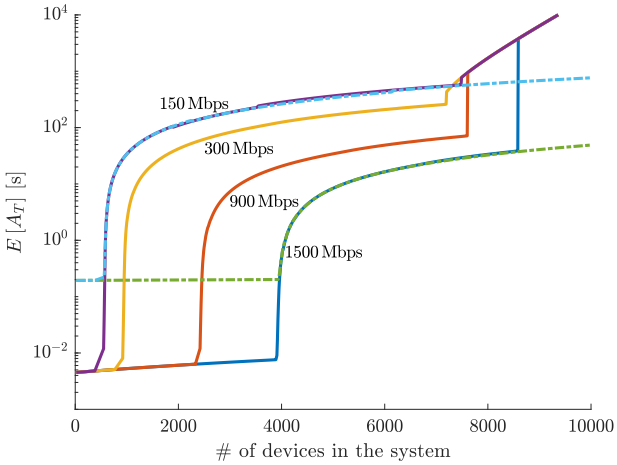

Figure 9: Access Delay for variable cell capacity without ACB (solid lines; 4 cases with capacity 150, 300, 900 and 1500 Mbps), and with ACB [$p_a = 0.95, E[B_a] = 4$] (2 dashed lines corresponding to extreme case capacities equal to 150 and 1500 Mbps).

It further fades exponentially with the access delay, normalized to the access delay value at n'. Thus, this second metric is very sensitive to relative increases of delay rather than to absolute increases.

The curves of the QoE parameters show qualitatively similar trends. As regards η_S, with a low population of UEs, the network access time $E[T_A]$ is very low and mostly depends on RACH transit and ACB operation. Each device in service is guaranteed a rate equal to R, keeping η_S close to 1, unless ACB is used and $E[A_T]$ cannot be neglected. When the BS can no longer provide the maximum rate R to each one of the n_S devices in service, $E[S]$ starts to increase, while $E[A_T]$ is practically constant (without ACB) or slowly increasing (with ACB), so that its weigh in η_S diminishes as the population increases. However, when the number of devices reaches the value n', the access delay $E[A_T]$ starts increasing fast (and linearly) causing a hyperbolic decrease of η_S towards zero. The QoE parameter starts dropping around 550 devices in the cell. In general, the figure shows that using ACB *is detrimental* in terms of quality of experience in steady state conditions, especially with small populations, when the ACB delay is the most prominent component of the access delay.

For what concerns η_A, the figure shows that, without ACB, it starts from the value $10/0.75 = 13.33$. This is the ratio between the data rate cap for each individual device, and the data rate given by the BS to each user once the maximum number of users (200) is reached (150 Mb/s divided by 200 users means 0.75 Mb/s per user). With ACB, the additional delay due to barring decreases the initial value of η_A. In all cases, the curve stays close to the initial value as long as the access delay remains negligible, then it rapidly drops. Also in this case, the QoE parameter starts dropping around 500 devices. Note that this means that a coverage of the $50,000$ users in the stadium with good QoE would require about 100 cells, if each user carries just one device, 200 cells if each user carries two devices, and so on.

Of course, one possibility to improve performance is to use cells with higher capacity. In Fig. 9 we plot curves of $E[A_T]$ for cell capacities in the range 150-$1,500$ Mb/s. The critical element for QoE is given by the points where the access delay starts increasing significantly. This means about 550 devices with capacity 150 Mb/s and about $4,000$ devices with capacity $1,500$ Mb/s. The latter translates into 12 cells for $50,000$ devices, 25 in the case each spectator carries 2 devices.

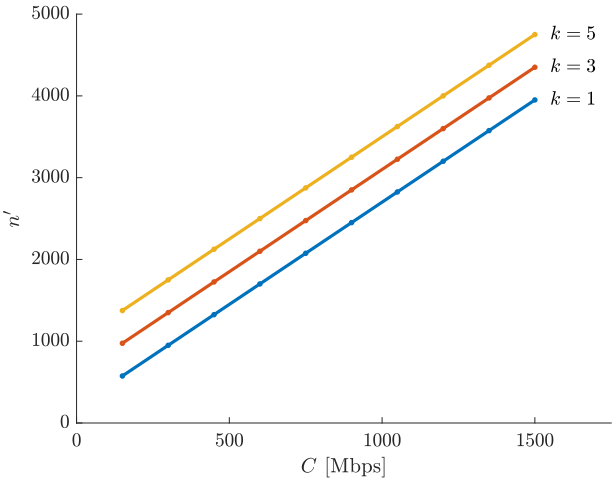

Figure 10: Values of n' versus the cell capacity, for variable cluster sizes. ACB curves are practically superposed to curves without ACB.

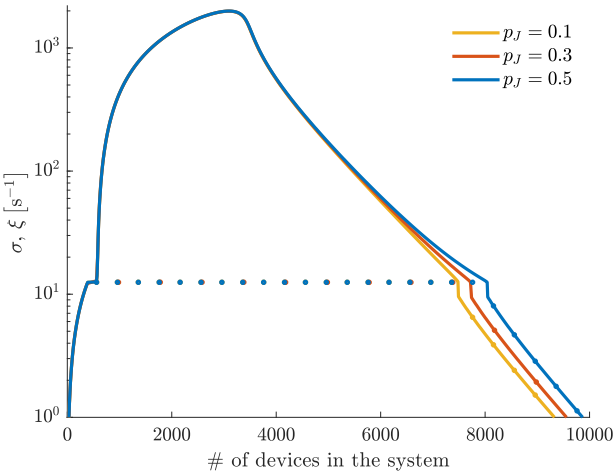

Figure 11: Impact of skipping the contention-based RACH procedure, σ and ξ solid and dashed lines respectively.

In addition, Fig. 9 clearly shows that the operating area where both end-users and network operators wish "to be" is just before the curve's first knee. In such neighbourhood, ACB does not play any significant role, and $E[A_T]$ is a fraction of $E[S]$, before starting to rapidly move to bigger values. It is important to recall that this "change of phase" in the access delay is pinpointed by n'. The second knee of the curves corresponds to n'', and both knees change with the cell capacity. It is very important to notice that in the whole interval $[n', n'']$ the system bottleneck is the Network due to the limitation of M `RRC_CONNECTED` devices. When the number of devices in the cell becomes larger than n'', we see a switch in the bottlenecks, and only from this point on the RACH subsystem becomes unstable and the access time explodes, going asymptotically to infinity.

Increasing the cell capacity or the number of cells is quite costly, and may not be the most desirable solution to achieve good QoE in crowded environments. A much simpler option can be to allow users to coalesce in their network access attempts through the formation of clusters. Fig. 10 shows the values of n' as a function of the cell capacity, for variable cluster sizes. We immediately appreciate the advantages of clusters: the adoption of coalitions brings a gain comparable to the one obtained increasing C with a negligible cost (if any) to the network provider. Indeed, a gain equal to or larger than that obtained by doubling the cell capacity can be achieved by adopting a cluster size $k = 3$.

Finally, to evaluate the importance of reducing the load of the Random Access in presence of downlink traffic, we repeat our tests with different values of p_J. Skipping the contention-based RACH procedure introduces a small improvement in terms of the height of the point at n^*, allowing the RACH to sustain a slightly higher arrival frequency. However, as can be seen in Fig. 11 for the case with no ACB, the main impact of p_J on the performance of the system is reflected in the value of n'', which is moved towards larger values of n. It must be noted that the increase in height at n^* is so small not to be visible on the graphs, and that the increase in the value of n'' is not relevant from the point of view of applications, because at those numbers of users per cell, performance (e.g., in terms of access delay) is intolerably bad.

7. Practical Validity of the Model and its Limitations

The analysis presented in this paper accounts for the main parameters of a 3GPP-compliant wireless

Table 2: Configuration parameters used to evaluate the practicality of our simple model (with $p_a = p_j = 0$)

Quantity	Notation	Value
Max number of RACH attempts	$k_{\max}$	$\{10, 20, 30\}$
Timeout of RRC_CONNECTED	RRC_{TO}	$\{1, 2, 3, 5, 10\}$ $[s]$
Application-related timeout (patience)	AR_{TO}	$\{15, 30, 60\}$ $[ms]$

access network. However, there are some parameters in the configuration of the network and in the application generating data to transmit that are not accounted for in the model and that are object of research and technical proposals [14, 15]. Those parameters can impose practical limitations to the behavior of the cellular access system. In particular, there are three main parameters that might be relevant and are not considered in the simple model presented and discussed so far: (*i*) the maximum number or RACH retries that a request can go through before being dropped; (*ii*) the timeout used in RRC_CONNECTED state, which guarantees that resources allocated to a device remain available beyond the last packet is transmitted, and which is useful to avoid incurring in a new RACH request procedure for customers returning within a few seconds; and (*iii*) the *patience* of a user, i.e., the maximum delay allowed by the application running at the user's device after which a transmission request is dropped and a new request is issued after some application-specific backoff.

In the following we analyze and discuss the impact of each of such parameters by means of simulations, whose results are compared to the model predictions. The configurations parameters used in what follows are summarized in Table 2. As we will see, our model captures the behavior of the system in a wide spectrum of configurations and, most important, it always gives accurate results for the regions of operation (i.e., the ranges of the number of users) that have practical importance.

7.1. Impact of the maximum number of retries on the RACH

Let's denote the maximum number of RACH retries as $k_{\max}$. This is the maximum number of consecutive failed RACH attempts, after which an access request is dropped. If such drop occurs, the device that was not granted network access will either give up or retry after some time (depending on the application), according to a backoff mechanism that has a length comparable with the think time of our model (tens of seconds). We therefore simulate a network in which, after a request fails RACH access for $k_{\max}$ consecutive times, the device goes back to the Think station of Fig. 1 without sending any data. Therefore, in this modified system, differently from the one used in the performance analysis of Section 6, it is possible to have *failures*. Note also that, for $k_{\max} \to \infty$, the system tends to the one we have modeled so far.

Fig. 12 shows that the impact of $k_{\max}$ on the throughput of the RACH (σ) and on the throughput of the Network station (ξ) is important only for population sizes $n > n^*$. Specifically, the smaller $k_{\max}$, the larger the distance $n'' - n'$ because n'' increases while n' does not change significantly. Similarly, n^* remains practically unchanged. This can be explained by considering that the probability to fail $k_{\max}$ consecutive times on the RACH is an increasing function of the load γ, and for populations below n^* devices, the probability to fail even as few as four or five times in a row is low because the probability of a single RACH failure is of the order of $1 - 1/e$ or smaller. The figure also shows that ξ does not change before n''.

Let's now consider the impact of $k_{\max}$ on the access delay. As depicted in Fig. 13, $E[A_T]$ is impacted only starting from the value n^* identified with our model. For larger values of the population of devices, the access delay decreases, still it remains quite high. This delay reduction is however obtained at the expense of the success probability experienced by a network access request. Indeed, as shown in Fig. 14, the system suffers high failure probability starting from $n = n^*$, which is the point at which the simple model shows a peak in the RACH throughput σ.

By jointly considering access delay and failure probability, it is clear that the network cannot be efficiently and satisfactorily operated with populations much larger than n'. As a consequence, our model offers accurate predictions well beyond the range of interest, since it is accurate up to $n^* > n'$.

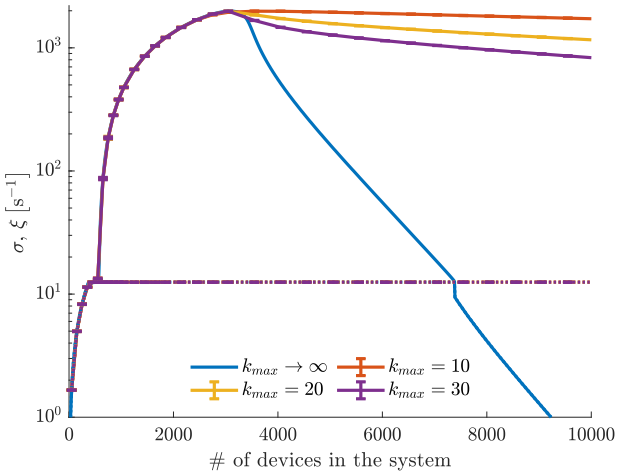

Figure 12: Effect of the max number of RACH retries $k_{\max}$ on σ and ξ, solid and dashed lines respectively

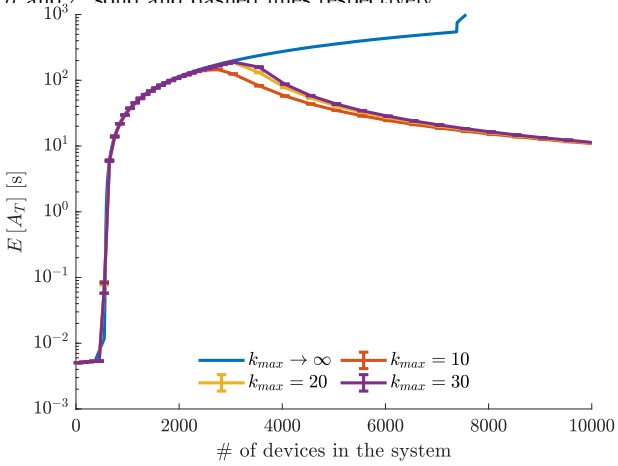

Figure 13: Effect of the max number of RACH retries $k_{\max}$ on the access time

7.2. Impact of timeout for the `RRC_CONNECTED` state

We now consider that, in real 3GPP access networks, devices that enter the `RRC_CONNECTED` state, leave that state based on an activity timeout, i.e., only after a time RRC_{TO} has elapsed, during which no transmission occurred. Otherwise, if after a file transmission is complete a device wants to initiate a new file transmission before the timeout expires, that device will not need to go through the RACH again. We simulate such system by (i) counting all devices in `RRC_CONNECTED` state, including the ones that have

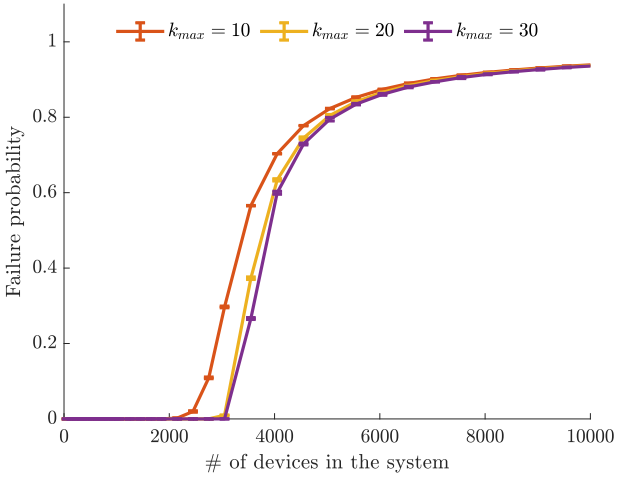

Figure 14: Failure probability introduced by the max number of RACH retries.

transmitted their file, in the set of devices that are under service, and whose number cannot exceed M; (ii) if a device exits the Think station and it is still in `RRC_CONNECTED` state, it jumps to the Network station of our system of Fig. 1. Note that our simple model corresponds to the case in which $RRC_{TO} = 0$. Note also that, differently from the case of $k_{\max}$ discussed before, the use of a timeout for the `RRC_CONNECTED` state does not lead to failures, although it allows less devices to use the transmission channel at the same time, as commented in what follows.

In this case, the network throughput ξ is practically not affected, as shown in Fig. 15, except n'' shifts forward due to returning devices sustaining ξ even when σ fades off. The Network in our system is PS, which means that when less than M devices are under service, they still use all resources, i.e., they are served faster than with exactly M devices. Therefore, although the use of a timeout $RRC_{TO} > 0$ imposes that some devices in the Think station do not free their Network allocation before RRC_{TO} time units after their file is transmitted, Network resources are not wasted. Fig. 15 also shows that the throughput of the RACH with $RRC_{TO} > 0$ is barely affected by the timeout. However, the longer the timeout, the higher and the sooner σ grows before n', and the later it falls after n''. This is due to the fact that if devices return to the Network station before the timeout expires,

the load of the RACH is alleviated, with less collisions experienced. However, in the interval between n' and n'', σ is simply shifted backward. Indeed, denoting by r the probability that the timeout does not expire, when Network is saturated the RACH sees a system with $M(1-r)$ serving slots instead of M, and our model can be applied to compute the resulting RACH throughput σ.

For what concerns access delay, Fig. 16 shows that only minor differences with respect to our model can be appreciated. Specifically, apart for a backward shift of n' and a forward shift of n'' that we have commented above, the linear slope of $E[A_T]$ in between is only slightly changed because the value of λ to be used in this case in Equation (8) is $\xi(1-r)$ instead of ξ. This effect is due to the increased probability to fail Network access, where rM allocation slots are now reserved for returning customers, and r increases with RRC_{TO}. This is only partially compensated by the short access delay experienced by devices for which the timeout does not expire, resulting in less delay at n' and higher delays at n''. In the extreme case in which $RRC_{TO} \to \infty$, the access delay is minimized for returning customers (it reduces to the latency of the Network station), although it becomes infinite for the rest of users, so that, as soon as $n > M$, the average access delay diverges.

Fig. 17 shows the average number of devices under service as a function of RRC_{TO}. The ratio between M and the value plotted in the figure is the coefficient r discussed above, which grows with the timeout and causes increased access delays.

In practical circumstances in which RRC_{TO} is of the same order of magnitude as the Think time, the probability r covers a small percentage of cases, and our model provides a good approximation for all values of the population n.

7.3. Impact of the application timeout

Finally, we consider the impact of the application timeout. We simulate the system of Fig. 1, although we interrupt and drop a service request if its access delay exceeds the application timeout AR_{TO}. Our simple model corresponds to the case $AR_{TO} \to \infty$. When a request is dropped, a failure occurs and the device goes back to the Think station.

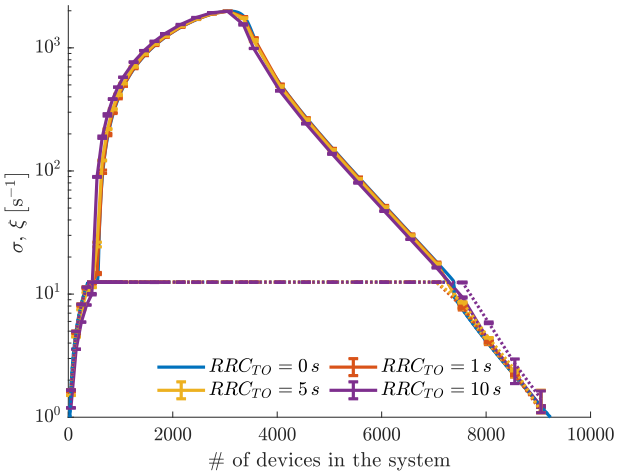

Figure 15: Effect of several configuration of RRC_{TO} on σ and ξ, solid and dashed lines respectively.

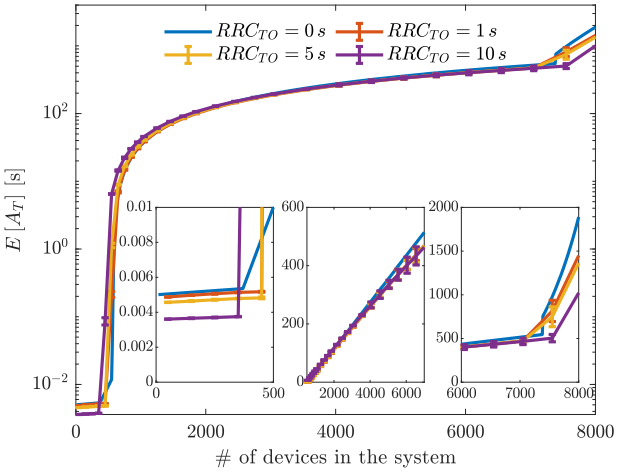

Figure 16: Effect of several configuration of RRC_{TO} on the access time, using a semi-log scale. The zooms represent $E[A_T]$ in linear scale, for populations in the ranges around n', within the interval $[n', n'']$ and around n'', respectively.

Fig. 18 compares the throughput of our model with the one of the simulator using various values of AR_{TO}. The figure unveils that no differences can be appreciated in σ and ξ for population size up to n' and slightly above that point. Beyond that point, the curves of σ separate. Both n^* and n'' increase with AR_{TO} decreasing. To explain this behavior,

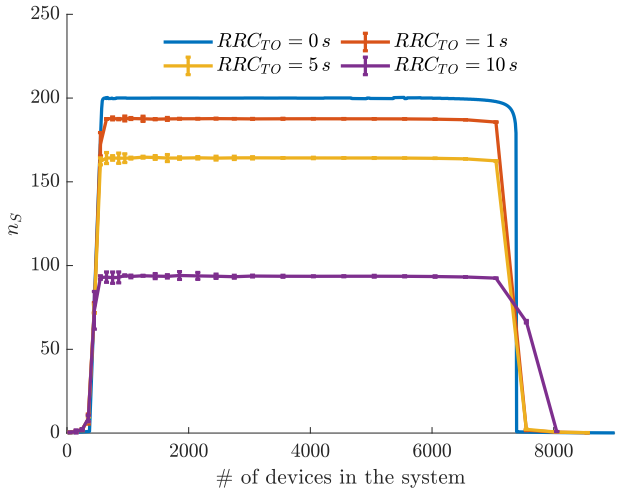

Figure 17: Number of users in service due to the use of a finite and not null timeout for the `RRC_CONNECTED` state.

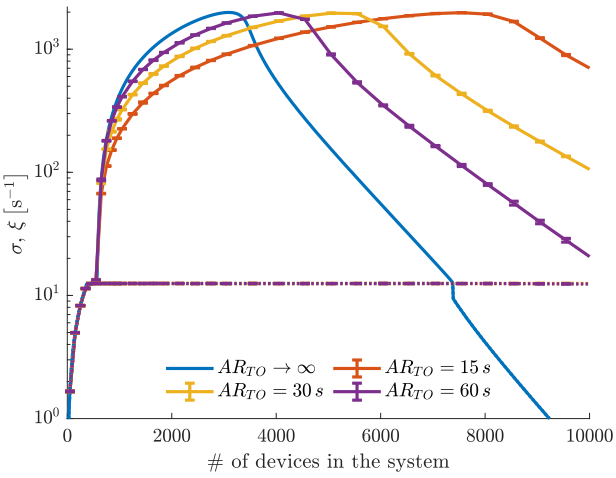

Figure 18: Effect of several configuration of the application timeout on σ and ξ, solid and dashed lines respectively.

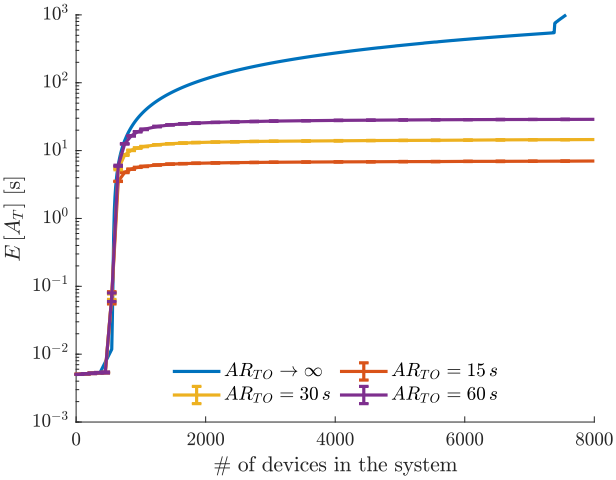

Figure 19: Effect of several configuration of application timeout on the access time

consider that the average access delay in our model increases with the population size. So, as soon as the average access delay increases, the probability to trigger an application timeout increases. With AR_{TO} larger than the RACH latency (in the order of τ, which is extremely small for an application timeout), a non-negligible timeout probability is possible only when the RACH experiences significant collisions, i.e., starting with some value of n between n' and n^*, which is what we observe in the figure. From that point on, requests that suffer timeouts take a Think time backoff, which is longer than the RACH or the Network backoff, thus resulting in reduced RACH load. This is why n^* and n'' move upwards.

For what concerns the impact on access delay, Fig. 19 shows significant differences from the point at which timeouts start occurring. Interestingly, the access delay with AR_{TO} grows very slowly after the curves in the figure split, and this is due to the hard bound imposed by the timeout. Therefore, using an application timeout could allow to use the system well beyond n' and even beyond n^*. However, as shown in Fig. 20, the failure probability becomes relevant well before n^*. So we conclude that using the system with populations much higher than n' is not a good idea even in this case.

In conclusion, the predictions of our model are very accurate for the range of population sizes in which the failure probability is negligible or bearable (below a few percentage).

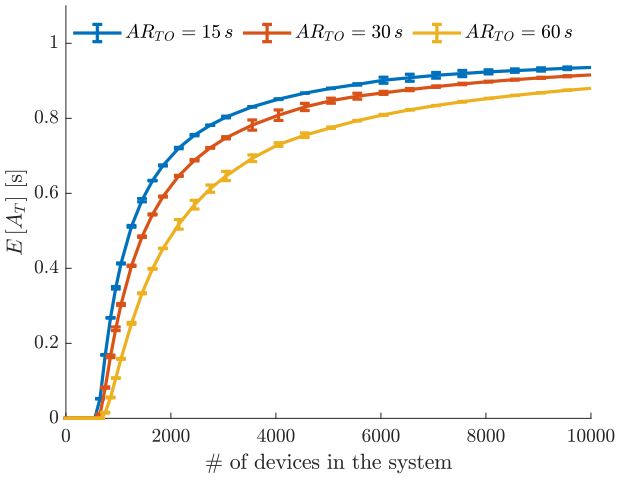

Figure 20: Failure probability introduced by the application timeout.

8. Related Work

In [16], 3GPP has identified the random access mechanism as a possible problem when the number of connected devices rises to tens of thousands. For this reason, MAC overload control has been investigated, and a broad literature exists on this topic. See [17] for a comprehensive overview. Simple models to estimate the probability of preamble collision in the PRACH channel are presented in a few 3GPP standard documents (e.g., [16]), and in the literature (e.g. [13], [17], [18], [19]). The conclusions of most of these studies point out that for Machine-Type Communications (MTC) applications, the Random Access procedure can drastically limit network performance. Possible approaches to modify the PRACH access procedure have been proposed in [20, 21, 22].

Most of the previous studies on dense cellular environments have focused on MTC scenarios, and [17] shows that the differences between the human-based and the MTC scenarios are substantial. Nevertheless, the PRACH access mechanism, and its interactions with the other phases of the network usage cycle play an important role also in case of human-based scenarios. This was shown in [1], through a measurement-based study of cellular network performance during crowded events, showing that network access failures become orders of magnitude higher than those observed on routine days, and the interaction between access and transmission phases generates behaviors difficult to predict. Recent works have shed light on the characteristics of the load of the Random Access subsystem. For instance, the authors of [23] derived the joint distribution of collisions and successes, so to be able to estimate the number of users based on observed events. Before that, the authors of [24] have shown how to model RACH successes in the presence of bursty arrivals. In [25], the authors propose a model of Random Access system with limits on the number of acknowledged requests per RAO, which is somehow equivalent to introduce network capacity constraints, although in a much simplified and rough manner. The simple analytical model presented in this paper is oblivious to traffic statistics and goes beyond previous models because it provides a tool to understand the root causes of the behaviors measured, e.g., in [1], and to quantify the impact of the crowd size on *RACH and network* performance as a tandem system. Moreover, our analysis permitted us to identify possible approaches to correctly dimension the network and—with the help of D2D sidelinks and BS-orchestrated clusters—to mitigate the negative impacts of crowds.

ACB and its extensions have been proposed and analyzed beyond MTC and crowded scenarios, so to be able to provide class-based priority in the Random Access [23, 26, 27]. These mechanisms promise to re-shape the RACH traffic load and even to adapt ACB parameters dynamically. However, this kind of approaches can only impact per-class delay statistics, not steady-state flows. Indeed, our model shows that the volume of flows at the entrance of the Random Access subsystem does not depend on the presence of ACB, see (1). Thus, ACB schemes are useful in scenarios in which the network operator needs to introduce priorities, but they do not improve system throughput.

Lately, radio access network limitations have become a hot topic, due to the start of deployment of 5G technology, where ultra-low latency and extremely dense scenarios are included in the standard operational framework. To cope with such challenging requirements, several proposals have been developed, such as for example reported in [28]. In order to reduce latency in 5G access networks, 3GPP

proposes a new unit of scheduling called a mini-slot, which can be flexibly configured to last between 1 and 6 orthogonal frequency-division multiplexed symbols. Furthermore, 5GPPP in [14] pinpoints group based RACH, that is coalescing access requests, as a solution aimed at handling the initial access bottlenecks due to massive connectivity.

9. Conclusions

This paper presents a model to capture the key aspects of the behaviour of cellular networks in crowded environments. The main merit of the model lies in the insight that it brings on cellular system operations in very crowded environments, and in the possibility to use it to drive the correct dimensioning of the cellular system in very crowded environments. As an example, the model allows the assessment of the benefits achievable through the adoption of D2D communications to reduce the congestion on the RACH more effectively than with ACB, thus significantly improving performance and QoE. For example, our model shows that, instead of serving 50,000 terminals with 100 cells of capacity 150 Mb/s each, it is possible to use 25 cells, each of capacity 300 Mb/s, provided that clusters of 5 devices are formed to access the RACH.

[1] M. Zubair Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, J. Wang, A First Look at Cellular Network Performance During Crowded Events, in: Proc. of the ACM SIGMETRICS '13, 2013, pp. 17–28.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022, Tech. rep. (Feb. 2019).
URL https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf

[3] W. Mohr, 5G empowering vertical industries, Tech. rep., Cisco (April 2016).
URL https://ec.europa.eu/digital-single-market/en/blog/5g-empowering-vertical-industries-0

[4] A. Asadi, Q. Wang, V. Mancuso, A Survey on Device-to-Device Communication in Cellular Networks, IEEE Communications Surveys & Tutorials.

[5] P. Castagno, V. Mancuso, M. Sereno, M. Ajmone Marsan, Why your smartphone doesn't work in very crowded environments, in: 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017, pp. 1–9. doi:10.1109/WoWMoM.2017.7974296.

[6] M. Höyhtyä, O. Apilo, M. Lasanen, Review of latest advances in 3gpp standardization: D2d communication in 5g systems and its energy consumption models, Future Internet 10 (1) (2018) 3.

[7] A. Asadi, V. Mancuso, R. Gupta, An SDR-based Experimental Study of Outband D2D Communications, in: Proc. of IEEE INFOCOM, 2016.

[8] Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification, TS 36.321 Release 13 V13.1.0, 3GPP (April 2016).

[9] S. Sesia, I. Toufik, M. Baker, LTE, The UMTS Long Term Evolution: From Theory to Practice, Wiley Publishing, 2009.

[10] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, V. Casares-Giner, Performance analysis of access class barring for handling massive M2M traffic in LTE-A networks, in: Proc. of IEEE ICC, 2016. doi:10.1109/ICC.2016.7510814.

[11] M. Ajmone Marsan, D. Roffinella, A. Murru, ALOHA and CSMA protocols for multichannel broadcast networks, in: Proc. of Canadian Commun. Energy Conf., Montreal, P.Q., Canada, 1982.

[12] R. Serfozo, Introduction to stochastic networks, Vol. 44, Springer Science & Business Media, 2012.

[13] J. J. Nielsen, D. M. Kim, G. C. Madueno, N. K. Pratas, P. Popovski, A Tractable Model of the LTE Access Reservation Procedure for Machine-Type Communications, in: Proc. of IEEE GLOBECOM, 2015.

[14] 5GPPP Architecture Working Group, View on 5G Architecture, Tech. rep., 5GPPP (December 2017).

[15] F. Alsewaidi, A. Doufexi, D. Kaleshi, A study on the influence of m2m gateways on the radio access channel of lte-a, in: 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2017, pp. 1–6. doi:10.1109/WCNCW.2017.7919083.

[16] Study on RAN Improvements for Machine-type Communications, TR 37.868 Release 11 V11.0.0, 3GPP (Septemeber 2011).

[17] L. A. Andres Laya, J. Alonso-Zarate, Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives, IEEE Communications Surveys & Tutorials 16 (1) (2011) 4,16.

[18] O. Arouk, A. Ksentini, General Model for RACH Procedure Performance Analysis, IEEE Communications Letters 20 (2) (2016) 372–375.

[19] G. C. Madueno, J. J. Nielsen, D. M. Kim, N. K. Pratas, C. Stefanovic, P. Popovski, Assessment of LTE Wireless Access for Monitoring of Energy Distribution in the Smart Grid, IEEE Journal on Selected Areas in Communications 34 (3) (2016) 675–688.

[20] T. P. C. de Andrade, C. A. Astudillo, N. L. S. da Fonseca, Random access mechanism for RAN overload control in LTE/LTE-A networks, in: Proc. of IEEE ICC, 2015.

[21] Y.-C. Yuan-Chi Pang, G.-Y. Lin, H.-Y. Wei, Context-Aware Dynamic Resource Allocation for Cellular M2M Communications, IEEE Internet of Things Journal 3 (3) (2016) 318–326.

[22] A. Grassi, G. Piro, G. Boggia, A look at random access for machine-type communications in 5th generation cellular networks, Internet Technology Letters 1 (1) (2018) e3.

[23] L. Tello-Oquendo, V. Pla, I. Leyva-Mayorga, J. Martinez-Bauset, V. Casares-Giner, L. Guijarro, Efficient random access channel evaluation and load estimation in lte-a with massive mtc, IEEE Transactions on Vehicular Technology 68 (2) (2019) 1998–2002. doi:10.1109/TVT.2018.2885333.

[24] C.-H. Wei, G. Bianchi, R.-G. Cheng, Modeling and analysis of random access channels with bursty arrivals in ofdma wireless networks, IEEE transactions on wireless communications 14 (4) (2015) 1940–1953.

[25] O. Arouk, A. Ksentini, T. Taleb, How accurate is the rach procedure model in lte and lte-a?, in: 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), 2016, pp. 61–66. doi:10.1109/IWCMC.2016.7577034.

[26] S. Duan, V. Shah-Mansouri, Z. Wang, V. W. S. Wong, D-acb: Adaptive congestion control algorithm for bursty m2m traffic in lte networks, IEEE Transactions on Vehicular Technology 65 (12) (2016) 9847–9861. doi:10.1109/TVT.2016.2527601.

[27] I. Leyva-Mayorga, M. A. Rodriguez-Hernandez, V. Pla, J. Martinez-Bauset, L. Tello-Oquendo, Adaptive access class barring for efficient mmtc, Computer Networks 149 (2019) 252 – 264. doi:https://doi.org/10.1016/j.comnet.2018.12.003.
URL http://www.sciencedirect.com/science/article/pii/S1389128618308211

[28] P. Popovski, J. J. Nielsen, C. Stefanovic, E. de Carvalho, E. Strom, K. F. Trillingsgaard, A.-S. Bana, D. M. Kim, R. Kotaba, J. Park, et al., Wireless access for ultra-reliable low-latency communication: Principles and building blocks, IEEE Network 32 (2) (2018) 16–23.

Paolo Castagno is Post-Doc at the Computer Science Department, University of Torino, Italy. He received his Master and Ph.D. degrees from the University of Torino, in 2014 and 2018, respectively. His research focus is on performance evaluation of computer systems and communication networks, with a specific interest on wireless networks.

Vincenzo Mancuso is Research Associate Professor at IMDEA Networks, Madrid, Spain, and recipient of a Ramon y Cajal research grant of the Spanish Ministry of Science and Innovation. Previously, he was with INRIA (France), Rice University (USA) and University of Palermo (Italy), from where he obtained his Ph.D. in 2005. His research focus is on analysis, design, and experimental evaluation of opportunistic wireless architectures and mobile broadband services.

Matteo Sereno was born in Nocera Inferiore, Italy. He received the Laurea degree in Computer Science from the University of Salerno, in 1987 and the Ph.D. degree in Computer Science from the University of Torino, in 1992. He is currently Full Professor at the Computer Science Department, University of Torino. His current research interests are in the area of performance evaluation of computer systems, communication networks, peer-to-peer systems, compressive sensing and coding techniques in distributed applications, game theory, queueing networks, and stochastic Petri net models.

Marco Ajmone Marsan is full professor at the Electronics and Telecommunications Department of the Politecnico di Torino in Italy, and part-time research professor at IMDEA Networks Institute in Leganes, Spain. Marco Ajmone Marsan obtained degrees in EE from the Politecnico di Torino in 1974 and the University of California, Los Angeles (UCLA) in 1978. He received a honorary doctoral degree from the Budapest University of Technology and Economics in 2002. Since 1974 he has been at Politecnico di Torino, in the different roles of an academic career, with an interruption from 1987 to 1990, when he was a full professor at the Computer Science Department of the University of Milan. Marco Ajmone Marsan has been doing research in the fields of digital transmission, distributed systems and networking. He has been a member of the editorial board and of the steering committee of the ACM/IEEE Transactions on Networking. He is a member of the editorial boards of the journals Computer Networksand Performance Evaluationof Elsevier, and of the ACM Transactions on Modeling and Performance Evaluation of Computer Systems. He served in the organizing committee of several leading networking conferences, and he was general chair of INFOCOM 2013. Marco Ajmone Marsan is a Fellow of the IEEE, a member of the Academy of Sciences of Torino, and a member of Academia Europaea.