Towards Random Access Channel Self-Tuning in LTE

Mehdi Amirijoo, Pål Frenger, Fredrik Gunnarsson, Johan Moe, Kristina Zetterberg
Wireless Access Networks, Ericsson Research, Ericsson AB, Sweden
{mehdi.amirijoo, pal.frenger, fredrik.gunnarsson, johan.moe, kristina.zetterberg}@ericsson.com

Abstract— Future radio access networks are expected to show a high degree of self-organization. This paper addresses self-tuning of the random access channel (RACH) in the 3G Long Term Evolution (LTE). The feasibility of self-tuning is investigated by means of simulation, where the coupling between several parameters and the performance of RACH is provided. The conclusion of the simulations is that RACH self-tuning is indeed possible given that UE assisted measurements are available for the self-tuning mechanism.

Keywords - LTE; Self-Organization; Random Access; E-UTRAN; Self-Tuning; Self-Optimization

I. INTRODUCTION

In parallel with the 3G Long Term Evolution (LTE) specification and development, the Next Generation Mobile Network (NGMN) association of operators brings forward requirements on management simplicity and cost efficiency [1][2]. The vision is that algorithms automate tasks that currently require significant planning efforts. One part of LTE that benefits from self-organization is the random access channel (RACH).

During initial access, the UE seeks access to the network in order to register and commence services. Optimal RACH performance is key to obtain high coverage and low delays, e.g., call setup delays, data resuming delays from the UL unsynchronized state, handover delays, and also to increase the physical uplink shared channel (PUSCH) capacity. Optimal setting of RACH parameters in LTE depends on a multitude of factors, e.g., the PUSCH interference from neighboring cells, cell size, RACH load, whether the cell is in high-speed mode or not (which affects the preamble detection probability), uplink (UL) and downlink (DL) imbalances, and the topology and the characteristics of the environment (e.g., urban and suburban). One approach is to use a set of standard parameter values in all base stations. This may, however, result in a suboptimal performance. Another approach is to – by means of simulation, prediction, or field trials – evaluate a wide range of random access parameters and choose those settings that satisfy given requirements.

The approaches given above are, however, not satisfactory since (1) there is a need to perform extensive simulations and field trials, (2) RACH parameters need to be reconfigured if network characteristics changes, e.g., the interference levels increases, or preambles need to be changed, UEs start moving in a higher speed in a cell (due to for example a highway built), and (3) finding a good set of parameter using simulation or field trials is a slow process and may not be sufficiently responsive to changes in network. As such, RACH may benefit from auto-tuning, where the network automatically sets RACH parameters based on observed characteristics and given performance requirements.

Self-organization and tuning have been previously addressed in the literature. For an overview on autonomic communication in networks refer to [3]. Automation of neighbor relation lists has received some attention lately [4][5]. Several publications related to automation in 3G networks exists, e.g., capacity and coverage balancing [6][7], and admission control [8]. The project SOCRATES aims at the development of self-organization methods for future wireless access networks [9]. Papers [18][19] address RACH optimization by means of simulations for WCDMA. To the best of our knowledge, however, there exists no previous work on automatic RACH parameter tuning.

The contributions of this paper are (i) a RACH performance specification model allowing operators or vendors to declare the desired or required performance, (ii) a study on the coupling and the degree of impact between tunable parameters and RACH performance, and (iii) a simple RACH power control self-tuning algorithm. For details regarding RACH in LTE we refer to appropriate literature (e.g., [10]) and standards (e.g., [11][12]).

The outline of this paper is as follows. A performance specification model is assumed in this paper. It may be likely that an operator or vendor would like to specify the probability that the UE has access after attempt \( m \) \((1 \leq m \leq M)\). We refer to this as the access probability \( AP_m \) at attempt \( m \). For example, the access probability should be greater than 0.9 and 0.9999 at attempts 1 and 3, respectively. There is a direct connection between the access probability and the access delay. Furthermore, the access probability \( AP_m \) is basically a function of two key factors, namely, the detection miss probability and the contention probability.

Define the detection miss probability \( DMP_m \) at attempt \( m \) as the probability of a preamble, transmitted at attempt \( m \), not being detected at the eNodeB. Further, let the \( AP_m \)
probability \( CP \) be the probability that a UE is not granted access due to a preamble collision, conditional that the preamble of the UE is detected. This gives an overall access probability at preamble transmission attempt \( m \),

\[
AP_m = 1 - \prod_{i=1}^{m} \left( DMP_i + (1 - DMP_i) \times CP \right).
\]

An alternative is to address the performance in terms of the detection miss probability \( DMP_m \) and the contention probability \( CP \). For the remainder of this paper we study the effects of various parameters on \( DMP_m \) and \( CP \), as \( AP_m \) depends on these probabilities.

### III. KEY PERFORMANCE METRICS

The following key performance metrics (KPMs) are formulated based on the performance specification model presented in Section II. Let \( T \) be the sampling period and \( y(k) \) denote the value of a variable \( y \) at time \( kT \). The following counters are used in the computation of the KPMs and are gathered over the time interval \( [(k-1)T,kT] \): number of sent preambles \( n_s(k) \), number of detected preambles \( n_d(k) \), and number of UEs that have succeeded with their access \( n_a(k) \).

An additional subscript \( m \) may be used to denote a particular attempt number. For example, \( n_{s,m}(k) \) gives the number of sent preambles for attempt number \( m \) during the time interval \( [(k-1)T,kT] \). The following RACH KPMs are considered in this report. Preamble detection miss ratio for attempt \( m \) is given by,

\[
DMP_m(k) = \begin{cases} 
1 - \frac{n_{d,m}(k)}{n_{s,m}(k)}, & n_{s,m}(k) > 0 \\
0, & n_{s,m}(k) = 0
\end{cases}
\]

The contention ratio is defined as,

\[
CR(k) = \begin{cases} 
1 - \frac{n_d(k)}{n_s(k)}, & n_s(k) > 0 \\
0, & n_s(k) = 0
\end{cases}
\]

\( DMR \) and \( CR \) are the estimates of \( DMP \) and \( CP \), respectively. To compute \( CR \) we need to measure the number of detected preambles \( n_d \) and number of UEs that are granted access \( n_a \). Both these metrics are directly measurable at the eNodeB and, as such, it is possible to measure \( CR \). To compute \( DMR \) we need to measure the number of sent preambles \( n_s \) and \( n_d \). Although it is possible to measure \( n_s \) it is not possible to measure \( n_d \) at the eNodeB unless this is reported by the UEs. An undetected preamble is simply a correlation peak below the detection threshold and is, therefore, classified as noise at the eNodeB detector. Henceforth, we assume that a UE reports the preamble of the UE, among the detected preambles. This gives the number of UEs that are granted access at attempt \( m \), which may then be used to estimate \( DMR_m \). Note, this particular UE report is not yet standardized at the moment of writing.

### IV. SIMULATOR OVERVIEW AND SCENARIO

The experiments presented in Section V are performed in a static simulator with time correlation. The length of a snapshot is one subframe (1ms) and the following steps are performed at each snapshot. First a set of UEs that are to perform a random access are created in this step, where the number of UEs created follows the Poisson process with a mean arrival intensity \( \text{Load}_{\text{PUSCH}} \) (number of UEs/second/cell). The generated UEs are uniformly distributed over the simulated area. UEs that succeed with their access are removed.

Next, the PUSCH load and the interference generated by PUSCH are simulated. Define the PUSCH load (denoted \( \text{Load}_{\text{PUSCH}} \)) as the ratio of resource blocks (RBs) that are scheduled for PUSCH to the total number of RBs available during a subframe (1 ms). Let RACH RBs denote those RBs where RACH is scheduled. We assume that the probability of scheduling PUSCH on RACH RBs is a linear function of \( \text{Load}_{\text{PUSCH}} \) hence, the probability of scheduling PUSCH on the RACH RBs increases with \( \text{Load}_{\text{PUSCH}} \). PUSCH power control is based on [13] and simplified to open-loop power control,

\[
P_{\text{PUSCH}} = \min\{P_{\text{max}}, P_{0,\text{PUSCH}} + PL\} \text{ dBm},
\]

where \( P_{0,\text{PUSCH}} \) is the desired target received power, \( PL \) is the path loss estimated by the UE based on the downlink reference signal, and \( P_{\text{max}} \) is the maximum transmission power.

The UEs execute random access consisting of the selection and the transmission of a preamble. Power control is based on [15] and modeled according to

\[
P_{\text{RACH}} = \min\{P_{\text{max}}, P_{0,\text{RACH}} - PL + (m - 1)\Delta_{\text{RACH}} + \Delta_{\text{Preamble}} + EE\}
\]

where \( P_{\text{RACH}} \) is the preamble transmit power, \( m \) is the RACH attempt number, \( P_{0,\text{RACH}} \) is the desired received power, \( \Delta_{\text{RACH}} \) is the power ramping step, \( \Delta_{\text{Preamble}} \) is a preamble-based offset, and \( EE \) models errors in PL estimation and transmission power. It is assumed that \( EE \) is i.i.d. normal random variable with zero mean and standard deviation \( \sigma_{EE} \).

The received preambles are processed, where the SINR of each preamble \( p \) received at cell \( c \) is computed according to,

\[
\text{SINR}_{p,c} = \frac{P_p g_{p,c}}{IP_{c} + N}
\]

where \( P_p \) is the transmission power of the UE transmitting preamble \( p \), \( g_{p,c} \) is the path gain from the UE to the eNodeB of cell \( c \), \( N \) is the thermal noise power over the RACH RBs, and \( IP_{c} \) is the received interference from PUSCH at cell \( c \). The SINR of each preamble is then mapped to a preamble detection probability (see [14]). If several UEs transmit the same preamble in a cell, then contention resolution is carried out by randomly choosing a preamble (UE) among the detected preambles.

The network is deployed in a hexagonal layout of 7 sites each 3-sectored and wrap-around propagation. Table I gives the value of parameters used in the simulations. For details regarding the antenna model and parameters we refer to [17].
such, interference. Consequently it seems that setting
experiments (if not otherwise stated).
by RACH. Table II gives the standard parameters used in all
of RACH (as defined in Section II), and the interference caused
coupling between various tunable parameters, the performance
indicates that there are a set of values for
The DMR of attempts 2-8 show similar behavior. Fig. 1(a)
very low
RACH. There may be cases where a
the models and assumptions of, e.g., propagation, PUSCH and
However, these results hold only for the deployment used and
to, e.g., -130 dB, will give a satisfactory RACH performance.

\[ L = 128.1 + 37.6 \log_{10}(d) \text{ [km]} \]

Log-normal shadowing 8 dB standard deviation

TABLE II. 

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoadPUSCH</td>
<td>0.5</td>
</tr>
<tr>
<td>LoadRACH</td>
<td>250 preambles/cell/s</td>
</tr>
<tr>
<td>RACH Format</td>
<td>0</td>
</tr>
<tr>
<td>RACH Configuration</td>
<td>6, 7, 8</td>
</tr>
<tr>
<td>P0RACH</td>
<td>-120 dB</td>
</tr>
<tr>
<td>ΔRACH</td>
<td>2 dB</td>
</tr>
<tr>
<td>M</td>
<td>8</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>180 s</td>
</tr>
</tbody>
</table>

V. EXPERIMENTS

The objectives of the simulations is to determine the
coupling between various tunable parameters, the performance
of RACH (as defined in Section II), and the interference caused
by RACH. Table II gives the standard parameters used in all
experiments (if not otherwise stated).

Note that the standard value for LoadRACH may seem too
high. Since at the time of the writing LTE has not been
deployed in large scale and typical loads are not yet available,
we assume a wide range of RACH loads in the simulations.
The default RACH load has been selected such that
\( \text{LoadRACH} = \{0, 2, 4, 6\} \text{ dB} \).

As shown in Fig. 1(a) the DMR of the first attempt (DMR1)
increases with increasing LoadPUSCH and decreasing P0RACH.
The DMR of attempts 2-8 show similar behavior. Fig. 1(a)
indicates that there are a set of values for P0RACH that result in
very low DMR and are robust to varying LoadPUSCH and, as
such, interference. Consequently it seems that setting P0RACH
to, e.g., -130 dB, will give a satisfactory RACH performance.
However, these results hold only for the deployment used and
the models and assumptions of, e.g., propagation, PUSCH and
RACH. There may be cases where a P0RACH lower (or higher)
than -130 dB should be used depending on prevailing conditions.

A. Effects of Varying PUSCH Load

The goal of this experiment is to study the effects of
P0RACH and LoadPUSCH on DMR and CR. Recall that P0RACH
dictates the received signal power and LoadPUSCH determines
the interference on RACH. The parameters are altered
according to LoadPUSCH = \{0.0, 0.2, ..., 1.0\} and P0RACH = \{-120, -130, -140, -150\} dB.

In conclusion, the PUSCH load and, thus, the PUSCH
interference, heavily affects DMR of all attempts. To counteract this, the power control parameter P0RACH can be
adjusted to an appropriate setting. Further, DMR and CR are
coupled, meaning that an increase in DMR results in an
increase in CR.

B. Effects of Varying Power Control Parameters

The goal of this experiment is to study the effects of
P0RACH and ΔRACH on DMR and to establish whether a given
performance specification (see Section II) in terms of DMR for
each attempt number can be satisfied. The parameters are altered according to P0RACH = \{-120, -130, -140, -150\} dB and
ΔRACH = \{0, 2, 4, 6\} dB.

The results are given in Fig. 2, where DMR of attempts 1, 3,
and 5 are given. In general, the DMR of all attempts decreases
nonlinearly with increasing P0RACH. As expected, for the first
attempt the DMR does not vary over ΔRACH. As such, the only
way to control DMR for the first attempt is to set P0RACH. For
attempts greater than one, DMR varies over both P0RACH and
ΔRACH. The amount by which DMR decreases when increasing
ΔRACH depends on the attempt number. This implies that for low
attempt numbers there are limits in how much we can alter
DMR using ΔRACH. As such, it may be necessary to alter P0RACH
to not only satisfy the first attempt, but also to satisfy attempt
numbers greater than one.
The conclusion of this experiment is that it is possible to control DMR by using $P_{0, RACH}$ and $\Delta_{RACH}$. The parameter $P_{0, RACH}$ can be set according to the DMR requirements of the first attempt, whereas $\Delta_{RACH}$ can be tuned to satisfy DMR requirements for the other attempts. In some cases the latter may not be possible and in such circumstances $P_{0, RACH}$ must be adjusted as well.

C. Effects of Varying RACH Load and Configuration

The goal of this experiment is to study the effects of RACH load and configuration on CR. RACH load is altered according to $Load_{RACH} = \{100, 300, \ldots, 900\}$ preambles/cell/s. RACH configuration is given by $\{(0, 1, 2), (6, 7, 8), (12, 13), 14\}$, resulting in random access opportunity periods of 20, 5, 2, and 1 ms. Note that $P_{0, RACH} = -120$ dB, which results in the majority of the preambles to be detected at the first attempt.

As expected CR increases with increasing $Load_{RACH}$ and increasing random access opportunity period (determined by the RACH configuration), as shown in Fig. 3. The conclusion of this experiment is that it is possible to control CR by altering the configuration.

D. Interference on PUSCH

The goal of this experiment is to study the interference on PUSCH generated by preamble transmissions. The idea is to show whether there is a benefit of adjusting $P_{0, RACH}$ in order to reduce the interference on PUSCH (compared to setting $P_{0, RACH} = -120$ dB). The generated interference is a function of the preamble transmission power and number preamble transmissions. For this reason we vary $P_{0, RACH}$ and the RACH load. The parameters are altered according to $Load_{RACH} = \{1, 5, 10, 25, 50, 100, 200, 300\}$ preambles/cell/s and $P_{0, RACH} = \{-120, -130, -140, -150\}$ dB. Define the PUSCH noise rise as,

$$NR = \frac{I_{RACH} + N_{RB}}{N_{RB}}$$

where $I_{RACH}$ denotes the received RACH inter-cell interference power (on PUSCH) and $N_{RB}$ is the noise power over one resource block (180 kHz). Assuming a receiver noise figure of 5 dB, we have that $N_{RB} \approx -146$ dBW.

The noise rise over all simulated cells is given in Fig. 4. The noise rise increases as a result of increasing $Load_{RACH}$ and $P_{0, RACH}$. The noise rise is substantial for some preamble transmissions when $P_{0, RACH} = -120$ dB and very small when $P_{0, RACH} = -140$ dB. Although, UEs close to the eNodeB may not suffer significantly from inter-cell RACH interference, UEs at the cell edge may, and the result may be a decreased PUSCH coverage and/or performance. As such, there may a benefit in decreasing the interference on PUSCH by lowering $P_{0, RACH}$.

The conclusion of this experiment is that RACH may cause interference on PUSCH for high $P_{0, RACH}$ and this may be alleviated by appropriately setting $P_{0, RACH}$ (given the performance requirements are satisfied).

E. Power Control Self-Tuning

The goal of this experiment is to illustrate that $DMR_1$ can be controlled to meet a given performance specification by automatically adjusting $P_{0, RACH}$. Due to lack of space we only show the results where $DMR_1$ is controlled and further extensions that include the control of other key performance metrics are left for future work. Recall from above that $DMR_1$ is heavily affected by PUSCH load. Therefore, we vary $Load_{PUSCH}$ according to Fig. 5. Although the stepwise changes in $Load_{PUSCH}$ may not be realistic, this gives the worst-case change in interference on RACH and, as such, allows us to study the performance of the controller under extreme conditions. The initial value of $P_{0, RACH}$ is -120 dB. Detection miss probability for the first attempt should be 0.01. An integrating controller (I controller)

$$P_{0, RACH}(k) = P_{0, RACH}(k-1) + K_i(0.01 - DMR_1(k))$$
is used where $K_I$ is a tunable parameter set to 0.6. The sampling period is set to 1s.

The results are given in Fig. 5, where the average over all cells is shown for $\text{Load}_{\text{RACH}}$, $P_{0,RACH}$, and $\text{Load}_{\text{PUSCH}}$. We can see that the controller is capable of adjusting $P_{0,RACH}$ such that the DMR1 tracks its target value (0.01). At time 45s and 80s, the PUSCH load increases significantly resulting in DMR1 overshoots. The overshoots cannot be avoided unless a mechanism that predicts the increase in $\text{Load}_{\text{PUSCH}}$ is available. We have in this experiment shown that using a simple I controller it is possible to control $P_{0,RACH}$ such that DMR1 satisfies a given performance specification in terms of a given target value.

VI. CONCLUSION

In this paper we have argued that there is an increasing need for self-organization in future wireless access networks. To meet these expectations there has been a significant effort carried out in academia, standardization bodies (e.g., 3GPP), and industry. One aspect that benefits from automation is RACH optimization. We have studied the feasibility of self-tuning of RACH in LTE by providing simulation results showing the impact of a set of key parameters on the RACH performance. Further an algorithm was presented, which tunes the RACH power control parameters such that the detection miss probability of transmitted preambles tracks given requirements. In order to automate the optimization of RACH power control parameters there is, however, a need for the UEs to report the number of sent preambles. These reports enable the derivation of preamble detection miss probability, which can be used for controlling the network access delay.

REFERENCES

[14] 3GPP R4-071951, “PRACH Simulation Results”, Ericsson