EUROPEAN COOPERATION
IN THE FIELD OF SCIENTIFIC
AND TECHNICAL RESEARCH

EURO-COST

COST 2100 TD(10)10056
Athens, Greece
2010/Febr/3-5

SOURCE: SOCRATES project consortium
c/o IBBT / University of Antwerp,
Antwerp, Belgium

Self-Optimising Call Admission Control for LTE Downlink

SOCRATES project consortium
c/o K. Spaey, B. Sas, C. Blondia
IBBT / University of Antwerp
Department of Mathematics and Computer Science
Middelheimlaan, 1
2020 Antwerp
BELGIUM
Phone: + 32-(0)3 265 38 80
Fax: + 32-(0)3 265 37 77
Email: {kathleen.spaey, bart.sas, chris.blondia}@ua.ac.be

Self-Optimising Call Admission Control for LTE Downlink

K. Spaey, B. Sas, and C. Blondia

Abstract— **The call admission control algorithm is the algorithm that decides if a call request will be admitted to the network or not. It bases its decisions on the availability of the resources needed to guarantee the required Quality-of-Service (QoS) of the new call, while at the same time maintaining the QoS of the already accepted calls. In this paper we focus on the self-optimisation of a call admission control algorithm for LTE downlink under sudden overload conditions. For this study, a reference admission control algorithm is considered, which distinguishes between handover and fresh calls based on a parameter denoted by Th_{HO}. A self-optimising algorithm which auto-tunes the parameter Th_{HO}, while pursuing an operator policy which prioritises the achievement of particular performance targets in case of conflicting targets, is defined. Simulation results illustrate that both before and after a sudden increase of the call arrival rate and the fraction of handover calls, the self-optimising admission control algorithm succeeds better in complying to the defined policy than the static algorithm with a fixed setting of Th_{HO}, for several fixed settings of Th_{HO}.**

Index Terms—**Call admission control, self-optimisation, LTE downlink.**

I. Introduction

As recognised by the standardisation body 3rd Generation Partnership Project (3GPP) [1] and the operator's lobby Next Generation Mobile Networks (NGMN) [2], [3], future wireless access networks, such as the 3GPP Long Term Evolution (LTE) radio access, will exhibit a significant degree of self-organisation. The principal objective of introducing Self-Organising Network (SON) functionalities in wireless access networks is to reduce the costs associated with network operations, while enhancing the network performance.

SON functionalities can be subdivided into 3 categories: (i) the self-optimisation of parameters, often also referred to as adaptive control or auto-tuning of parameters; (ii) self-healing tasks, which aim at minimising the impact of network failures on the network performance; and (iii) self-configuration, which enables the automatic derivation of sensible initial configurations for network equipment during the installation process.

In this paper, we focus on the self-optimisation of a call admission control algorithm for LTE downlink under sudden overload conditions. The call admission control algorithm is the algorithm that decides if a call request will be admitted to the network or not, based on the availability of the resources needed to guarantee the required Quality-of-Service (QoS) of the new call, while at the same time maintaining the QoS of the already accepted calls. In LTE, the admission control algorithm is hosted by the eNodeB, and thus typically operates autonomously on a per cell basis.

This paper has been written in the context of the SOCRATES project (www.fp7-socrates.eu), which is funded by the European Union within the 7th Framework Program and which runs from 2008-2010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the SOCRATES project.

K. Spaey, B. Sas and C. Blondia are with the Department of Mathematics and Computer Science, University of Antwerp - IBBT, Middelheimlaan 1, 2020 Antwerp, Belgium (e-mail: {kathleen.spaey, bart.sas, chris.blondia}@ua.ac.be).

A. Related Work

Admission control in wireless networks has been studied in many papers and for many technologies. Designing efficient admission control algorithms is a challenging task, because of the diverse QoS requirements for delay-sensitive and delay-tolerant applications, the heterogeneous traffic patterns of calls originating from different applications and the requirement to prioritise the acceptance of handover calls over fresh calls, as dropping an ongoing call that is handed over is considered more annoying to a user than the initial rejection of a fresh call.

Many admission control algorithms that deal with how to ensure that priority can be given to handover calls compared to fresh calls, are built upon a concept known in the literature as 'guard channels' [4], [5]. A subset of the available capacity, which is typically expressed as a number of available channels, is reserved for handover calls. These reserved channels are called guard channels and an important issue in algorithms considering guard channels is to choose the right number of guard channels. The algorithm developed in [4] is an adaptive algorithm which searches automatically for the optimal number of guard channels.

In [5]-[7], differentiated admission control for delay-tolerant and delay-sensitive traffic streams is considered. In [6], [7], the proposed admission control algorithm limits the fraction of the capacity that can be occupied by real-time calls. As such, the more delay-tolerant non-real-time traffic can be used as a 'buffer' that protects the real-time traffic which has more stringent delay requirements in case of temporary overload.

To the best of our knowledge, except for [6], papers considering admission control for cellular networks start from the assumption that the capacity of a cell is fixed. This way, the performance (call rejection probability) of an admission control algorithm can be evaluated by considering the admission control algorithm in isolation. However, in LTE the downlink cell capacity varies over time, due to the varying radio conditions at the user equipments (UEs) and the channel-aware scheduling decisions taken by the downlink scheduler. Therefore, in this paper we relax the assumption of a fixed cell capacity and take the method of [6] to estimate the time-varying cell capacity into account.

The remainder of this paper is organised as follows. In Section II, a reference admission control algorithm is defined and the method used to estimate the time-varying cell capacity is described. Section III gives an overview of the main characteristics of the simulation model, the performance metrics and measurements that are considered, and the evaluation methodology that will be used. Section IV defines the operator policy which deals with the trade-off between various performance objectives, and which is incorporated in the self-optimising algorithm defined in Section V. This algorithm auto-tunes Th_{HO}, which is one of the main parameters of the reference admission control algorithm. In Section VI, the simulation set-up is described and simulation results are presented and discussed. The paper is concluded in Section VII.

II. Reference Admission Control Algorithm

Based on admission control algorithms encountered in the literature, we have defined a reference admission control algorithm which accepts handover (HO) calls with priority over fresh calls, which distinguishes between fresh real-time (RT) and non-real-time (NRT) calls and which takes the time-varying cell capacity into account. The admission decision rules used by the reference algorithm are described in Section II.A, and the method used to estimate the time-varying cell capacity is presented in Section II.B.

A. Admission Decisions

The core of an admission control algorithm are the decision rules by which the algorithm decides if a call is admitted to the cell or not.

Consider a call request that arrives at time t, and suppose that upon time t, C(k) is the most recent estimate of the cell capacity (see Section II.B). Denote by c_{req} the required capacity of the arriving call, by $c^*(t)$ the total required capacity of the already accepted active calls, and by $c^*_{RT}(t)$ the total required capacity of the already accepted active real-time calls.

Figure 1 shows a flowchart of the reference algorithm. The algorithm has 4 parameters, Th_{HO}, Th_{RT}, Th_{MAX} and λ, which all take a value between 0 and 1, with Th_{HO}, Th_{RT} and λ smaller than or equal to Th_{MAX}. The parameter Th_{MAX} can ensure that the available capacity C(k) is not fully consumed, as a first safety margin to catch temporary congestion. The purpose of Th_{HO} is to prioritise handover calls. When a fresh call arrives, and the total needed capacity $c^*(t)+c_{req}$ is larger than Th_{HO} x C(k), the call is blocked. In this way, Th_{HO} makes sure that some capacity is reserved for handover calls. If an arriving fresh call passes the test which involves Th_{HO} and the call is a non-real-time call, it is accepted. For a real-time call, further tests are performed with the purpose to avoid that a too large part of the available cell capacity becomes filled with real-time calls, without some capacity being used by non-real-time calls. The motivation for this is the fact that real-time calls have more stringent delay requirements than non-real-time calls. A way to still satisfy the delay requirements of real-time calls in temporary overload situations is to exploit the delay-tolerant characteristic of the non-real-time traffic as an elastic 'buffer', by using a packet scheduler such as the one defined in [8] which takes packet urgencies into account. However, this is only possible if the cell capacity is not fully filled with real-time calls. This is tested in the reference algorithm by the test involving the parameter λ, in case more than a fraction Th_{RT} of the available cell capacity would be in use if the arriving fresh real-time call would be accepted.

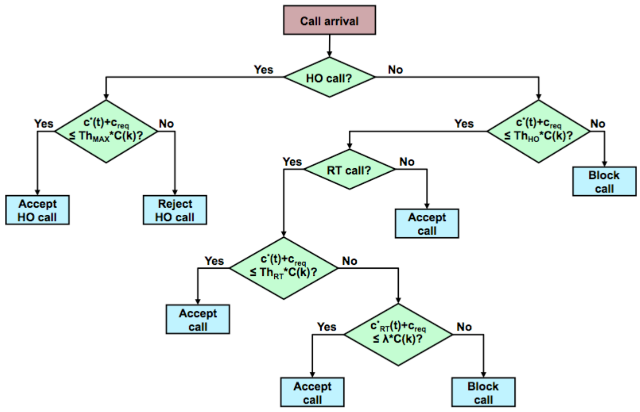

Figure 1: Flowchart of the reference admission control algorithm.

B. Estimation of Time-Varying Cell Capacity

In LTE downlink a channel is shared among multiple users using orthogonal frequency division multiple access (OFDMA). It is the packet scheduler which controls, at each scheduling period, to which users the available resources are assigned. The scheduling resources in LTE downlink have a granularity of 180 kHz in the frequency domain, and 1 ms (also called a transmission time interval (TTI)) in the time domain. To cope with the variable channel conditions and to optimise the overall data rate, LTE uses adaptive modulation and coding. So as a consequence of the decisions taken by the scheduler and the ability in LTE to adjust the bit rates to the varying channel conditions, the total bit rate at which data can be transmitted every scheduling time varies dynamically over time.

Based on the method proposed in [6], we estimate the time-varying cell capacity by measuring the number of bits that can be transmitted in a certain time interval, according to the assignments of scheduling resources by the packet scheduler. Suppose that time is divided in fixed length intervals of T TTIs, so each interval lasts T ms. In each interval, a cell has M = T x C scheduling resources (see Figure 2) for which the packet scheduler decides which scheduling resource is assigned to which user. The bit rate at which data is sent towards the user to which a scheduling resource is assigned, depends on the used modulation and coding scheme. The total number of bits that are transmitted in interval k then equals

$$b(k) = \frac{T}{1000}\sum_{t=1}^{T}\sum_{c=1}^{C} r(k,t,c),$$

where r(k,t,c) denotes the bit rate (expressed in bits/s) at which the user to which the scheduling resource of the t-th TTI and c-th subchannel of interval k is assigned, is served. Note that it is possible that some of the M available scheduling resources in an interval remain idle. So when estimating the cell capacity from measurements of the number of transmitted bits in an interval, a correction factor that takes these idle scheduling resources into account is needed. Denote by m(k) the number of scheduling resources that are effectively used during interval k to transmit data, and by μ(k) the estimated throughput (expressed in bits/s) in interval k. Then

$$\mu(k) = b(k)\frac{M}{m(k)}\frac{1000}{T} = \frac{M}{m(k)}\sum_{t=1}^{T}\sum_{c=1}^{C} r(k,t,c).$$

The factor M/m(k) corrects the measured throughput for unused scheduling resources; in case there are no unused scheduling resources, this factor equals 1. The reference admission control algorithm then calculates C(k), the cell capacity at the end of interval k, using an exponentially weighted average, as

$$C(k) = (1-\alpha)\ C(k-1) + \alpha\ \mu(k).$$

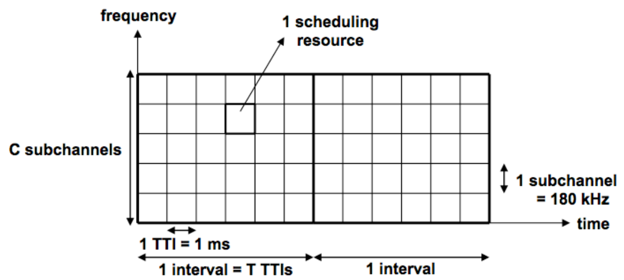

Figure 2: T x C scheduling resources corresponding with an interval.

III. Simulation Model, Performance Metrics and Evaluation Methodology

A dynamic system level simulator has been written for performing a sensitivity analysis on the parameters of the reference admission control algorithm and for studying the self-optimisation algorithm that will be presented in Section V. The main characteristics of this simulator are described in Section III.A. Section III.B discusses some performance metrics relevant for assessing a call admission control algorithm and the collection of measurements in the simulator. The methodology that will be used in Section VI to evaluate the performance of the self-optimising algorithm is presented in Section III.C.

A. Simulation Model

The structure of the simulator is shown in Figure 3. The simulator is developed using the OPNET Modeler [9] software and simulates the downlink direction. The reference admission control algorithm described in Section II, the packet scheduling algorithm of [8] and the self-optimising algorithm defined in Section V are implemented in the simulator. The main characteristics of the simulator are:

- Call generation. Calls are generated according to a Poisson process with a given call arrival rate expressed in calls/s. For simplicity, it is assumed that every UE can receive only one call at the same time. A fraction of the generated calls is considered to represent handover calls, the rest are fresh calls. The traffic type of a call (voice, video, web) is determined according to a chosen traffic mix.
- Call admission control. Every generated call is first offered to the admission control (AC) algorithm, which decides about its acceptance or rejection.
- Admitted calls. For every call accepted by the admission control algorithm, a traffic source corresponding to the type of the accepted call is initiated. Also a buffer in which packets destined for the UE associated with the call are queued is set up. Every TTI, the scheduling algorithm assigns the scheduling resources among the users that have packets available in their buffer. The aggregate bit rate over all users and the number of scheduling resources that remain idle during the TTI are fed back to the admission control algorithm, which uses this information to estimate the time-varying cell capacity as described in Section II.B. From the UEs, signal to interference and noise ratio (SINR) values are fed back to the scheduling algorithm to inform it about the link quality on every subchannel for every UE. The scheduling algorithm uses this information to decide about the modulation and coding schemes (MCSs) and corresponding bit rates to be used on the scheduling resources it assigns to the UEs.
- SON algorithm. If the self-optimising algorithm defined in Section V is activated in the simulator, it will collect performance measurements on the calls rejected by the admission control algorithm and on the QoS experienced by the traffic that is generated by the calls that are admitted into the network. Based on these measurements, the SON algorithm might decide on a change of the value of the parameter Th_{HO} of the admission control algorithm, in which case it will feedback this new value towards the admission control algorithm, which from that moment on will use the new value.
- Simulation scenario. The simulation scenario consists of a single cell with the eNodeB located in the middle. The eNodeB antenna is isotropic and the transmit power is equally distributed over all subchannels. A 5 MHz spectrum allocation (25 subchannels) is used.
- Traffic models. Three types of traffic are considered: voice traffic, video traffic and web traffic. Each call is dedicated to one of these three types. Voice calls and video calls are categorised as real-time calls, and web calls as non-real-time calls. To generate traffic of these types in the simulator, the traffic models and their corresponding parameter values as described in [10] are used. For the web browsing traffic, it is assumed that a web browsing session consists of a single packet call, which corresponds with the downloading of a single webpage composed of a main page and a number of embedded objects.
- Mobility model. As mobility model the random waypoint model is used. The start location of a UE depends on if the call that is associated with the UE is a fresh or a handover call: UEs corresponding to fresh calls are uniformly positioned in the entire cell, while UEs corresponding to a handover call are uniformly positioned on the cell border.

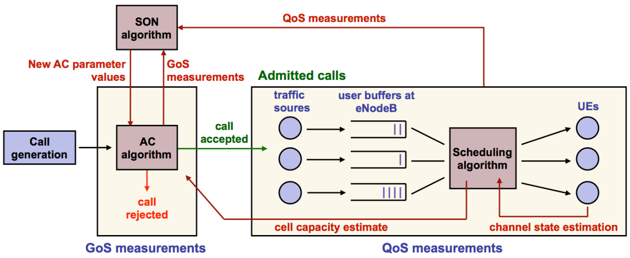

Figure 3: Structure of the developed simulator.

B. Performance Metrics and Measurements

It is expected that the key gains from employing a self-optimising admission control algorithm will be in the form of enhanced performance (Grade-of-Service (GoS) and QoS).

GoS metrics are metrics on the rejection or acceptance of calls by the admission control algorithm. As GoS metrics, the *call blocking ratio* and the *handover failure ratio* will be considered. The call blocking ratio is defined as the ratio of the number of fresh calls rejected by the admission control algorithm to the total number of fresh calls offered to the admission control algorithm, and the handover failure ratio is the ratio of the number of rejected handover calls to the total number of handover calls offered to the admission control algorithm. GoS measurements are collected in the simulator from the admission control algorithm.

QoS refers to performance associated with the traffic that is handled by the network, so to measurements on the performance experienced by the traffic that is generated by calls that are admitted into the network by the admission control algorithm. For the real-time traffic, the QoS is assessed by the *traffic loss ratio*. In the simulator, the only cause of traffic loss is when the time the traffic has spent in its eNodeB buffer exceeds the maximum allowed delay, as then the traffic will be dropped from the buffer as it will become useless to the receiver. For the results presented in this paper, a maximum allowed delay of 50 ms is assumed for voice traffic, and of 200 ms for video traffic. The QoS of the non-real-time web traffic is evaluated by considering the fraction of web calls with a *call throughput* smaller than the minimum call throughput requested to the packet scheduler (assumed to be 250 kbit/s). The call throughput is obtained by dividing the total number of bits associated with a call by the time needed to complete the call. In the simulator, QoS measurements are gathered from the part of the simulator that handles the traffic that is accepted by the admission control algorithm.

C. Evaluation Methodology

In Section VI, the performance of the self-optimising algorithm for Th_{HO} (see Section V) will be evaluated in a scenario that suddenly changes from a "normal" situation for which the network is dimensioned, into an overload situation for which the network is not dimensioned. Such situations typically occur when an unforeseen event happens, like a sudden announcement at an airport that for some reason all flights are temporarily suspended. In the simulator, such a situation is simulated by introducing at a certain moment in time a change in the call arrival rate and/or the fraction of handover calls, as is illustrated in Figure 4. The state the system is in before this change is called state A, that after the change state B.

GoS and QoS metrics will be collected over the period that the system is in state A, and over the time the system is in state B. To assess the performance of the self-optimising admission control algorithm which auto-tunes Th_{HO}, the results obtained with the self-optimising algorithm will be compared with the outcomes acquired with the static algorithm for several fixed values of Th_{HO}.

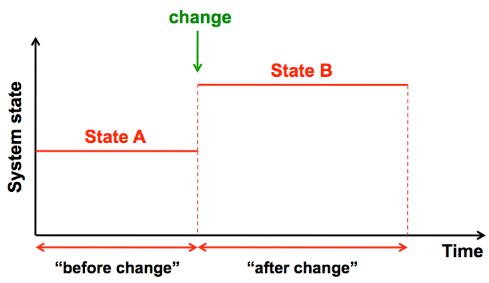

Figure 4: System state change.

IV. Operator Policy

In the next section, a self-optimising algorithm for Th_{HO} will be developed, which auto-tunes Th_{HO} based on QoS and GoS measurements. Changes in the measured performance might however require opposite adaptations of Th_{HO}, depending on which performance metric is considered. From performing a sensitivity analysis on the influence of Th_{HO} on the experienced GoS and QoS performance [11], it is known that an increasing call blocking ratio demands for an increase of Th_{HO}, while an increasing handover failure ratio will demand for a decrease of Th_{HO}. A QoS degradation of the ongoing calls will also demand for a decrease of Th_{HO}.

In principle, an operator may target the ideal goal of achieving the best possible QoS for the ongoing calls and the lowest possible rejection ratios for the fresh and the handover calls. In most cases, it will however not be possible that all these goals can be fulfilled at the same time. As a consequence, an operator policy is needed to decide on how the targets have to be weighted or ranked.

In this paper, we consider the following operator policy, where the items are ranked in order of priority:

1) The first aim should be to guarantee the QoS of the already accepted calls;
2) The second aim should be to have a low handover failure ratio;
3) The third aim should be to have a low call blocking ratio.

V. Self-Optimising Algorithm for Th_{HO}

From the sensitivity analysis on the influence of Th_{HO} on the experienced GoS and QoS performance [11], it has been concluded that for call arrival rates and fractions of handover calls that generate a load for which the network is dimensioned, the value of Th_{HO} does not matter much for the QoS of the accepted calls and for the handover failure ratio. Th_{HO} might however have a significant influence on the call blocking ratio of the fresh calls: the lower Th_{HO} is set, the higher the call blocking ratio becomes. So it is advantageous for the fresh calls to set Th_{HO} to a high value. However, when the call arrival rate and/or the fraction of handover calls grows, the impact of Th_{HO} on the QoS of the accepted calls and on the handover failure ratio also increases. So if the admission control algorithm wants to pursue the operator policy defined in Section IV, the safest option is to set Th_{HO} to the largest value suitable for inferior system conditions. But then, under "normal" conditions, many fresh calls will be rejected unnecessarily, meaning that the revenue of the network provider will be limited needlessly. Therefore, it would be much better if Th_{HO} could be auto-tuned to changes in the system conditions, based on observations of the QoS and GoS performance.

Consider the following self-optimising algorithm for Th_{HO}. At regular time instants t = kΔ, with k an integer and Δ a time interval, the following measurements are collected: M_QoS_RT(k), M_QoS_NRT(k), M_GoS_HO(k), and M_GoS_fresh(k), which are defined as respectively the traffic loss ratio of the real-time traffic, the fraction of non-real-time calls for which the call throughput is smaller than the minimum call throughput requested to the packet scheduler, the handover failure ratio and the call blocking ratio, in the time interval [kΔ;(k+1)Δ[. Note that all these measurements will result in a value between 0 and 1.

As there will be fluctuations in the measurements, they should be filtered such that the self-optimising algorithm reacts on time-averaged measurements rather than on instantaneous observations. Using an exponentially weighted average with parameter α_{SON} results in the filtered measurement QoS_RT(k):

$$\text{QoS\_RT}(k) = (1-\alpha_{SON})\ \text{QoS\_RT}(k-1) + \alpha_{SON}\ \text{M\_QoS\_RT}(k).$$

In a similar way, also QoS_NRT(k), GoS_HO(k) and GoS_fresh(k) are obtained from the respective measurements M_QoS_NRT(k), M_GoS_HO(k) and M_GoS_fresh(k).

Pseudo-code for the self-optimising algorithm for Th_{HO} is shown in Figure 5. τ_{QoS_RT}, τ_{QoS_NRT}, τ_{GoS_HO} and τ_{GoS_fresh} are threshold values with which the filtered measurements are compared, and when they are exceeded, a change of Th_{HO} might happen. The threshold values should be chosen between 0 and 1. Note from Figure 5, lines 1-4, that Th_{HO} will be lowered if the filtered measurement value of the QoS of the real-time calls, or of the QoS of the non-real-time calls, or of the handover failure ratio exceeds the corresponding threshold, i.e., if a bad QoS or handover failure ratio is experienced. If all these values are lower than 90% of their corresponding thresholds, and the rejection ratio of the fresh calls is larger than τ_{GoS_fresh}, then Th_{HO} will be increased (lines 6-10). The motivation for increasing Th_{HO} comes from a too high rejection ratio of the fresh calls. However, by allowing such an increase only if the QoS of the ongoing calls and the handover failure ratio are below 90% of their thresholds, and by taking smaller steps for increasing Th_{HO} than for decreasing it, the policy described in Section IV to first aim at a good QoS and low handover failure ratio before targeting a low call blocking ratio is pursued.

At time t = (k+1) Δ, execute:

```
1  if ( QoS_RT(k) > τ_QoS_RT
2        OR QoS_NRT(k) > τ_QoS_NRT
3        OR GoS_HO(k) > τ_GoS_HO )
4    Th_HO = max(Th_HO - 0.1 , 0);
5  else
6    if ( QoS_RT(k) ≤ 0.9 τ_QoS_RT
7          AND QoS_NRT(k) ≤ 0.9 τ_QoS_NRT
8          AND GoS_HO(k) ≤ 0.9 τ_GoS_HO
9          AND GoS_fresh(k) > τ_GoS_fresh )
10     Th_HO = min(Th_HO + 0.05 , 1);
11   end
12 end
```

Figure 5: Pseudo-code for the self-optimising algorithm for Th_{HO}.

VI. Simulation Set-up, Results and Discussion

As mentioned in Section III.C, the self-optimising admission control algorithm will be evaluated by comparing performance results obtained by applying the self-optimising algorithm in a scenario where at a certain moment in time a change in the system conditions is introduced, with the results obtained by applying the admission control algorithm with a fixed setting of Th_{HO} in the same scenario. Results from both before and after the change will be compared. Section VI.A discusses the simulation set-up, while Section VI.B presents and discusses the obtained results.

A. Simulation Set-Up

The following set-up is considered for the simulations presented in this paper:

- During each simulation run, 3000 calls are generated. For each simulation with identical parameters, 10 independent simulation runs are run.
- The traffic mix consists of 1/3rd voice calls, 1/3rd video calls and 1/3rd web calls, with a mean holding time for the voice and video calls of 90 s.
- A scenario where at a certain moment in time a change in the call arrival rate and the fraction of handover calls is introduced, is considered. While the system is in state A (see Figure 4), calls are generated at a rate of 0.6 calls/s, and 30% of these calls are considered to be handover calls. After 28 minutes, i.e., after approximately 1000 calls have been generated, the call arrival rate is increased to 1 call/s and 60% of the calls are considered to be handover calls.
- The static admission control algorithm is applied for several fixed settings of Th_{HO}: Th_{HO} = 0.3, 0.4, ..., 1. If the self-optimising admission control algorithm is applied, Th_{HO} is initialised to 0.8, and afterwards auto-tuned. Results with the self-optimising algorithm are obtained for two values of the exponential smoothing

parameter α_{SON}, which is used to filter the measurements the algorithm needs as input.

- The parameter values used for the parameters of the reference admission control algorithm and of the self-optimising algorithm for Th_{HO} are set as specified in Table 1. Note that the parameters Th_{MAX}, Th_{RT} and λ of the admission control algorithm are set equal to 1, such that only Th_{HO} influences the obtained results.

Parameter	Parameter value	Parameter meaning
α_{SON}	0.75, 0.90	Exponential smoothing parameter used to filter the measurements the self-optimising algorithm uses as input
τ_{QoS_RT}	1e-5	Threshold values to which the filtered measurements used by the self-optimising algorithm are compared
τ_{QoS_NRT}	2%	
τ_{GoS_HO}	1%	
τ_{GoS_fresh}	5%	
Δ	1 minute	Time interval after which the self-optimising algorithm reconsiders if an update of Th_{HO} is needed
Th_{MAX}	1	Parameters of the reference admission control algorithm (see Figure 1)
Th_{RT}	1	
λ	1	
α	0.1	Exponential smoothing parameter used when estimating the time-varying cell capacity
T	5	Number of TTIs in the time intervals used to estimate the time-varying cell capacity
c_{req}, voice call	6.10 kbit/s	Required capacity of a call, taken into account by the admission control algorithm
c_{req}, video call	64 kbit/s	
c_{req}, web call	250 kbit/s	

Table 1: Parameter values used for the parameters of the self-optimising algorithm and the reference algorithm.

B. Simulation Results and Discussion

In this section, results obtained with the self-optimising admission control algorithm are indicated with 'SON', and results obtained with the static algorithm get the label 'no-SON', both in the figures and in the text. Figure 6 shows the results on the handover failure ratio and Figure 7 on the call blocking ratio of the fresh calls. Figure 8 presents results on the fraction of web calls with a call throughput smaller than 250 kbit/s, i.e., QoS results for the non-real-time calls, and Figure 9 shows the traffic loss ratios of the voice and video traffic, which are the QoS results for the real-time traffic.

From these figures it is seen that:

- Before the change, with the no-SON algorithm, a good QoS for the ongoing calls and a low handover failure ratio is obtained for all fixed settings of Th_{HO}. The SON cases perform equally well when considering these performance measures before the change. Also with respect to the call blocking ratio, the SON cases result in good performance before the change, while for the no-SON cases this is only the case for high settings of the Th_{HO} value ($Th_{HO} \geq 0.8$). So when putting these results together, it is concluded that with the static algorithm, before the change the best performance is obtained with a high Th_{HO} value. The SON algorithm performs equally well or slightly worse (call blocking ratio) than these best no-SON cases. The reason that SON performs sometimes slightly worse is because before the SON algorithm will change Th_{HO}, first some bad performance needs to be measured, and of course this bad performance is also included in the results.
- After the change, with the no-SON algorithm, the best QoS and handover failure ratio results are obtained with $Th_{HO} \leq 0.6$. Again, the SON algorithm succeeds in obtaining equally good or slightly worse (for the same reason as explained before) results for these performance measures than these best no-SON results. When looking at the call blocking ratio results after the change, it is seen that the no-SON cases with a high setting of Th_{HO} give the best results. However, it were exactly these no-SON cases which gave the worst results for the QoS and the handover failure ratio after the change. The no-SON cases with smaller Th_{HO} settings which resulted in the best QoS and handover failure ratio, give the worst call blocking ratio results. The SON algorithm gives better call blocking ratio results than the no-SON cases with $Th_{HO} \leq 0.6$, and worse results than the no-SON cases with higher Th_{HO} values. As according to the policy defined in Section IV, a good QoS of the ongoing calls and a low handover failure ratio should get priority over a low call blocking ratio, the results illustrate that this is indeed what the SON algorithm pursues, while this is impossible to achieve with the static algorithm with a high Th_{HO} value. The no-SON algorithm with a lower Th_{HO} value also pursues the operator policy, but worse performance is obtained with these no-SON cases than with the SON algorithm, both before and after the change, with respect to the call blocking ratio.
- SON results have been collected with α_{SON} = 0.75 and α_{SON} = 0.90. The obtained results with both values for α_{SON} turn out to be rather similar, the main difference is noticed in the results for the call blocking ratio after the change. The higher α_{SON} value of 0.90 seems better in this case.

The results discussed above illustrate that overall, the presented self-optimising admission control algorithm succeeds better in complying to the defined operator policy, both before and after the change, than the static algorithm with a fixed Th_{HO} value. The reason for this is that a fixed setting of Th_{HO} which results in good compliance of the measured performance to the operator policy when the system is in a specific state (i.e., a certain mix of handover/fresh calls and a certain call arrival rate), is likely to lead to poor results when the system is in another state. With the self-optimising algorithm, because this algorithm auto-tunes Th_{HO} based on performance measurements, Th_{HO}

can evolve to a new good value after changes to the incoming traffic.

In [11], we investigated some more scenarios, in which a change corresponded to an increase of the call arrival rate only, or of the fraction of handover calls only. Similar conclusions as for the results presented in this paper were drawn for these scenarios.

VII. Conclusions

In this paper, based on properties of admission control algorithms encountered in the literature, first a reference admission control algorithm has been defined. Then a simple self-optimising algorithm has been proposed, which auto-tunes Th_{HO}, one of the main parameters of the defined reference algorithm, while pursuing an operator policy which prioritises the achievement of particular performance targets in case of conflicting targets. Through simulation results it has been demonstrated that before and after a sudden increase of the call arrival rate and the fraction of handover calls, the self-optimising algorithm succeeds better in complying to the defined operator policy than the static algorithm with a fixed Th_{HO} value. The reason is that a fixed setting of Th_{HO} which is optimal when the system is in one specific state corresponding to a certain mix of handover/fresh calls and a certain call arrival rate, might not be optimal for another system state. With the self-optimising algorithm, because this algorithm auto-tunes Th_{HO} based on performance measurements, Th_{HO} can evolve to a new good value after a state change. When the system is in one specific state, the proposed self-optimising algorithm might perform slightly worse than the static algorithm with an optimal Th_{HO} value corresponding to that state, because the self-optimising algorithm needs to measure some bad performance first before it will trigger an adaptation of the Th_{HO} value.

References

[1] *Telecommunication Management; Study on Management of Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC)*, 3GPP TR 32.816 (Release 8) v8.0.0, Dec. 2008.

[2] *Next Generation Mobile Networks Beyond HSPA & EVDO*, NGMN White Paper, Dec. 2006. Available: http://www.ngmn.org.

[3] *NGMN Use Cases related to Self Organising Network, Overall Description*, NGMN Deliverable, May 2007. Available: http://www.ngmn.org.

[4] Y. Zhang and D. Liu, "An adaptive algorithm for call admission control in wireless networks", in *Proc. Globecom 2001 Conf.*, San Antonio, TX, 2001, pp. 3628-3632.

[5] W. Ahmad, I. Awan, "Performance evaluation of CAC schemes for multi-service traffic environment in 3G networks", *SIMULATION,* vol. 83, issue 3, pp. 217-227, March 2007.

[6] S.S. Jeong, J.A. Han and W.S. Jeon, "Adaptive connection admission control scheme for high data rate mobile networks", in *Proc. VTC 2005 - Fall Conf.*, Dallas, TX, 2005, pp. 2607-2611.

[7] H. Lei, M. Yu, A. Zhao, Y. Chang, D. Yang, "Adaptive connection admission control algorithm for LTE systems", in *Proc. VTC 2008 - Spring Conf.*, Sigapore, 2008, pp. 2336-2340.

[8] I. Fernandez Diaz, R. Litjens, J.L. van den Berg, D.C. Dimitrova, and K. Spaey, "Sensitivity of the optimal parameter settings for a LTE packet scheduler", accepted in *Proc. VTC 2010 - Spring Conf.*, Taipei, Taiwan, May 2010.

[9] OPNET Modeler http://www.opnet.com/solutions/network_rd/modeler.html.

[10] *NGMN Radio Access Perfomance Evaluation Methodology*, NGMN white Paper, Jan. 2008. Available: http://www.ngmn.org.

[11] K. Spaey, B. Sas, N. Van den Wijngaert *et al.*, "Self-simulation methods for stand-alone functionalities in wireless access networks: admission control parameter optimisation", SOCRATES deliverable D3.1B, Dec. 2009.

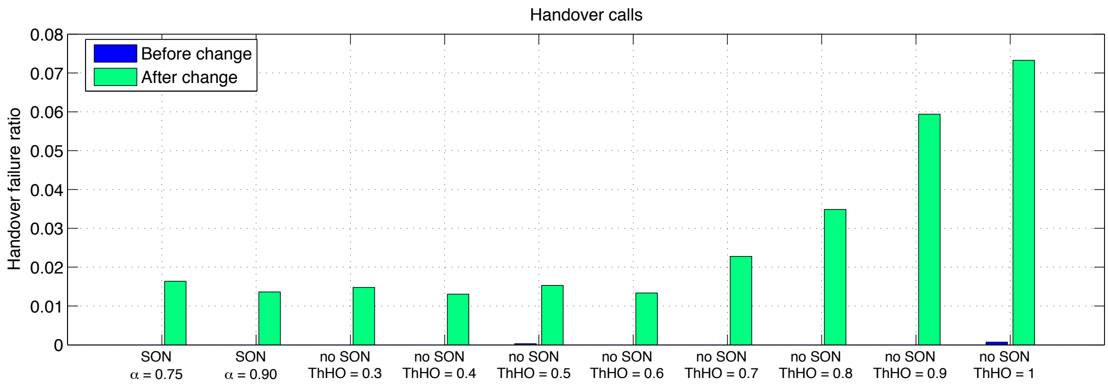

Figure 6: Handover failure ratios (GoS results), with and without self-optimisation.

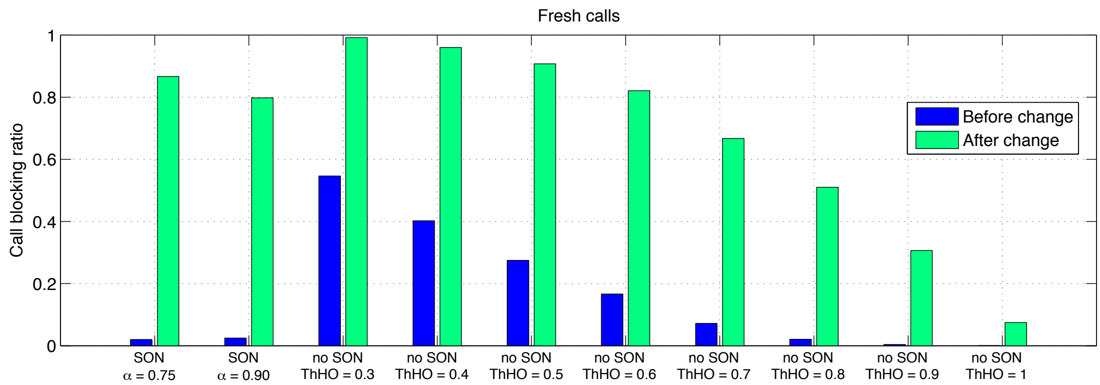

Figure 7: Call blocking ratios (GoS results), with and without self-optimisation.

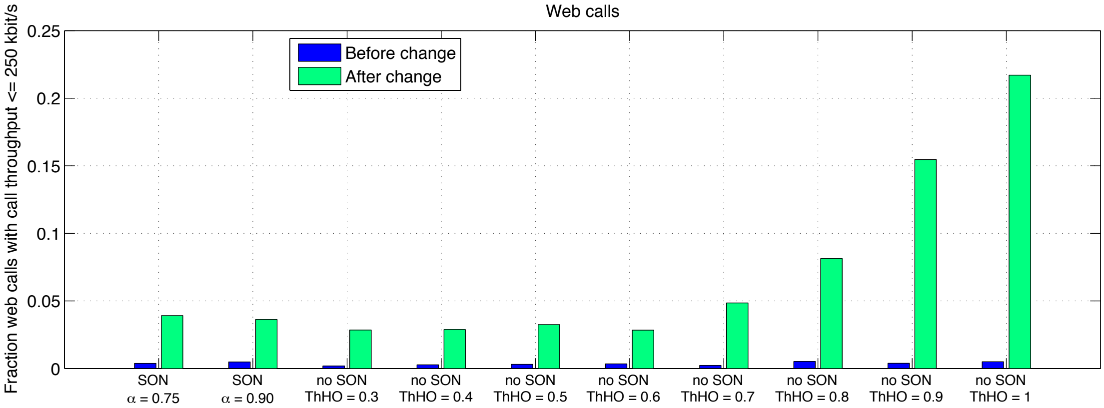

Figure 8: Fractions of web calls with a call throughput smaller than 250 kbit/s (QoS results), with and without self-optimisation.

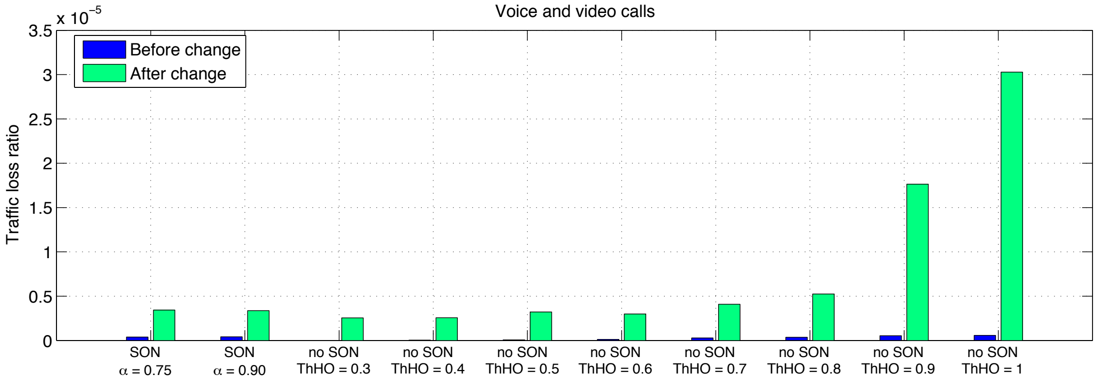

Figure 9: Traffic loss ratios (QoS results), with and without self-optimisation.