



HAL
open science

Congestion Control in the context of Machine Type Communications in Long Term Evolution networks

Ahmed Amokrane

► **To cite this version:**

Ahmed Amokrane. Congestion Control in the context of Machine Type Communications in Long Term Evolution networks. Networking and Internet Architecture [cs.NI]. 2011. dumas-00636132

HAL Id: dumas-00636132

<https://dumas.ccsd.cnrs.fr/dumas-00636132>

Submitted on 26 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ENS CACHAN, BRITANNY EXTENSION
UNIVERSITY OF RENNES I

RESEARCH MASTER IN COMPUTER SCIENCE,
NETWORKS AND DISTRIBUTED SYSTEMS

MASTER THESIS INTERNSHIP REPORT

Congestion Control in the context of Machine Type Communications in Long Term Evolution networks

Work done by :
Ahmed Amokrane
ahmed.amokrane@ens-cachan.fr

Supervisors:
Adlen Ksentini
adlen.ksentini@irisa.fr
Yassine Hadjadj-Aoul
yassine.hadjadj-aoul@irisa.fr

DIONYSOS TEAM
IRISA / INRIA RENNES BRETAGNE ATLANTIQUE

2010/2011

Abstract

Machine Type Communications (MTC) are automated applications which involve machines or devices that communicate through a network without any human intervention. They can be used today in almost everyday life applications from military to civil applications, such as: transportation, health care, smart energy, supply and provisioning, city automation... Those devices are generally spread in a wide area and should communicate through widely deployed networks. A good candidate to play the role of such a network could be cellular mobile networks. In fact, Cellular networks have been already deployed and offer a large coverage. Such a deployment is beneficial for both mobile network Operators (i.e. more revenues) and the application developers (i.e. more opportunities). Furthermore, LTE networks are all-IP networks and offer a good support for MTC. However, cellular networks are not designed for MTC applications. Consequently, such a deployment is challenging and rises new problems. The most important of them is congestion. In fact, in MTC, a huge number of devices are deployed. This leads to contention and congestion in the the different parts and nodes of the network when a lot of devices communicate at the same time. In the present work, we propose a novel Congestion-Aware Admission Control solution, which deals with the congestion in LTE networks. Our solution, which is based on control theory, effectively allows avoiding the core network congestion while saving the wireless scarce resources. We show its effectiveness through network simulations carried on the ns-3 simulator. In fact, the obtained results show that our solution is robust against different traffic patterns and accommodates huge amount of devices as expected in this particular case of MTC applications. It assures a targeted utilization of the resources in the network. Furthermore, it is completely compliant with the actual protocols and standards.

Keywords: MTC, LTE, PID, Congestion Control, Admission Control.

Contents

1	Introduction	4
2	Background and related works	5
2.1	Overview of Machine Type Communications	5
2.2	MTC applications taxonomy	6
2.3	Overview of the Long Term Evolution (LTE) networks	7
2.4	MTC applications in LTE networks	8
2.5	Overview of PID controllers	9
2.6	The congestion problem	12
2.7	Related works	13
3	Our proposal: “Congestion-Aware Admission Control”	14
3.1	Introduction	14
3.2	Congestion metrics	16
3.3	Congestion-Aware Admission Control (CAAC)	17
4	Performance evaluation	19
4.1	Simulation model	19
4.2	Test scenarios	20
4.2.1	Bursty traffic	21
4.2.2	Uniform traffic	22
4.2.3	Uniform traffic with bursts	27
4.2.4	Random traffic	30
4.3	Discussion	33
5	Discussions on our proposed solution	34
5.1	Advantages	34
5.1.1	Impacts on the existing nodes and 3GPP standards	34
5.1.2	Enhancement of several proposed approaches	34
5.1.3	Flexibility of the congestion metric	35
5.1.4	Protection against starvation	35
5.1.5	Impacts on the different parts of the LTE networkss	35
5.1.6	Benefits for Mobile Network Operators	36
5.2	Weaknesses and possible improvements	36
6	Conclusion	37
	References	38

List of Figures

1	A general architecture of a M2M application	6
2	The LTE Global architecture	7
3	Sequence Diagram for signaling and data in case of Event Driven and Time Driven applications	9
4	Sequence Diagram for signaling and data in case of Query Driven applications	9
5	PID controller, general diagram	10
6	The impact of the gains (K_p , K_i and K_d) on the output value [1]	11
7	The congestion problem in LTE network in the context of MTC applications	12
8	LTE Network Architecture	15
9	General Overview of the Solution	18
10	LTE Network Testing Topology	21
11	The Signaling traffic generated by the MTC Devices in case of bursty traffic	22
12	The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Bursty traffic	23
13	The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Bursty traffic	23
14	Comparison between the case where a PID queue and drop tail queue are used in the MME in the case of Bursty traffic	24
15	The Signaling traffic generated by the MTC Devices in case of Uniform traffic	25
16	The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Bursty traffic	26
17	The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Uniform traffic	26
18	Comparison between the case where a PID queue and drop tail queue are used in the MME in case of Uniform traffic	27
19	The Signaling traffic generated by the MTC Devices in case of Uniform traffic with bursts	28
20	The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Uniform traffic with bursts	29
21	The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Uniform traffic with bursts	29
22	Comparison between the case where a PID queue and drop tail queue are used in the MME in case of Uniform traffic with bursts	30
23	The Signaling traffic generated by the MTC Devices in case of Random traffic	31
24	The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Uniform traffic with bursts	32
25	The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Random traffic	32
26	Comparison between the case where a PID queue and drop tail queue are used in the MME in case of Random traffic	33

1 Introduction

Machine Type Communications (MTC) (also called Machine-to-Machine (M2M)) are automated applications which involve machines or devices communication through a network without any human intervention. They can be used today in almost all everyday life applications from military to civil applications, such as: transportation, health care, smart energy, supply and provisioning, city automation, and manufacturing. The communicating devices can be embedded in different environments like cars, cell towers, vending machines...etc. They are generally spread in a wide area and should communicate through widely deployed networks. A good candidate to play the role of such networks could be the cellular mobile networks.

Cellular mobile networks offer different network technologies for M2M communications. There are strong realistic predictions stating that M2M will be leveraged over cellular mobile networks. In fact, such integration represent many advantages for both Mobile Network Operators (MNO) and application developers. It will provide MNO with more revenues and the applications developers with more opportunities. Therefore, efforts are being conducted to enable such deployment. The 3rd Generation Partnership Project (3GPP)¹ [2] is working on specifications to standardize the deployment of M2M applications in 3GPP networks (UMTS and LTE). However it is challenging and not trivial. In fact, cellular networks (among them LTE) are designed for Human-to-Human (H2H), Human-to-Machine (H2M) and Machine-to-Human (M2H) applications, which are different from M2M. Thus, MNO should accommodate their networks to support the M2M applications which involve a huge amount of autonomous devices. One of the most important problems posed by that deployment is congestion. Congestion concerns all the parts of the network, both the radio and the Core Network. It is due, mainly, to the fact that MTC devices are deployed in a huge number and are generally synchronized (more likely to send data at the same time). This may lead to peak load situations and may have a tremendous impact on the operations of the mobile network, penalizing both MTC and non-MTC devices. For instance, if many devices detect an event at the same time, they send their alerts toward a central server at the same time, leading to congestion in the different nodes of the network.

In the present work, we propose a novel solution to handle the congestion in LTE networks in the particular context of MTC applications. Our solution is implemented at the network level (IP) and doesn't impose any constraint on the applications. It combines the actions of both a Congestion Control and an Admission Control: it uses an queue monitoring based on a Proportional, Derivative and Integrative (PID) controller implemented in each Core Network node to detect congestion while enabling Admission Control which rejects traffic in the radio part. Practically, the core network nodes monitor congestion and define the parameters of the Admission Control in the radio part.

Furthermore, we give some numerical results on the simulations performed under the ns-3 [3] simulator. The results concern the context of MTC applications through traffic load and patterns that are more likely to be those of MTC. We show the effectiveness of our proposal through the obtained results. In fact, we succeeded to reduce the amount of signaling, to reach a targeted

¹3GPP is a collaboration between groups of telecommunications associations that aim to standardize the 3G and the 4G Mobile systems to come through LTE and LTE advanced. It is composed of European Telecommunications Standards Institute (ETSI), Association of Radio Industries and Businesses/Telecommunication Technology Committee (ARIB/TTC, Japan), China Communications Standards Association (CCSA), Alliance for Telecommunications Industry Solutions (ATIS, North America) and Telecommunications Technology Association (TTA, South Korea).

utilization of resources in the core network part and to avoid losses (overflow in the queues). Note that our proposal is completely compliant with the existing standards and existing architectures.

The remainder of this document is organized as follows. First, Section 2 presents a gentle overview of M2M, LTE networks and PID controllers. The problem of congestion as well as the related works are, also, presented in this section. Section 3 presents a general and detailed description of our proposed solution. The proposed solution is, then, evaluated in Section 4 through simulations and discussions on the results. Then, Section 5 discusses our proposed solution. Finally, Section 6 concludes the document.

2 Background and related works

In this section, we briefly present some generalities on MTC, LTE networks and the Proportional, Integrative and Derivative (PID) controller we are using. Then the congestion problem we are tackling is presented as well as the related works in literature.

2.1 Overview of Machine Type Communications

MTC or M2M is, usually, a form of data communication which involves one or more entities that do not necessarily need human intervention [4]. In a practical scenario, a common M2M application uses a *device* (sensor, meter...) to capture an event (temperature, inventory level, etc.) by converting analog measurements to digital data, more precisely IP packets. The digital data is sent through a network (wireless, wired or hybrid, UMTS, LTE..., depending mainly on the required QoS [5] and the cost) to an application (software program) called a *server*². The latter translates it into meaningful information (data stored, threats detected...) [6, 7]. Afterwards, answers could be sent back to the devices.

The devices involved in a M2M application and having the required functionalities, consisting in replying to requests and/or sending data by themselves, are called *M2M devices* (or *MTC devices*) [4].

A generic and simple architecture of a MTC application is illustrated in Figure 1. The architecture is mainly composed of mainly three parts: *M2M Device Domain* containing the MTC devices, *Network Domain* which transports the messages between the MTC devices and the servers located in the *Application Domain*, giving data to business applications the devices are deployed to. The network domain could be UMTS or LTE as we will see it in details later.

MTC applications are different from H2H, M2H and H2M applications. In fact, they should behave and assume their roles without any human intervention. Furthermore, they are characterized by some properties that make them a bit more different. The main properties are [4]:

- Large number of devices: Typical applications require a lot of devices deployed in the same area leading to high density or distant leading to a large spread.
- Devices can send and/or receive frequently or infrequently small amounts of data (low data rates).

²Note that some other practical scenarios may involve only devices.

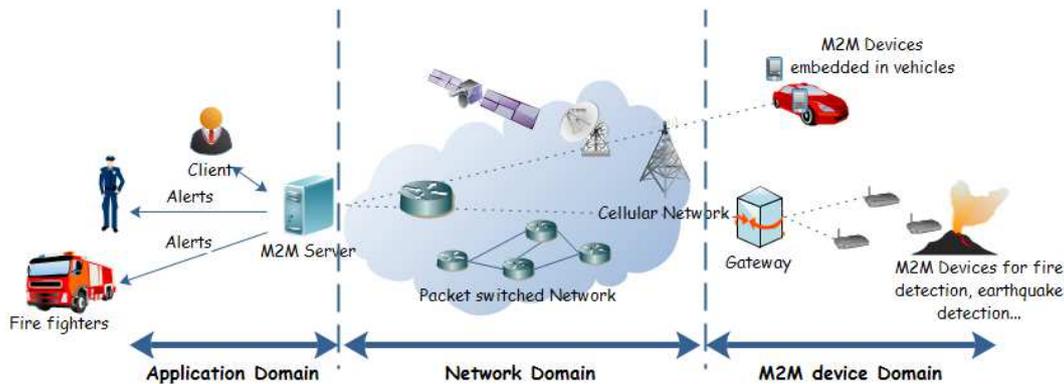


Figure 1: A general architecture of a M2M application

- Low mobility for devices meaning that they don't move, move infrequently or move in a predefined region.
- Reduced costs as these applications are used in everyday life and may involve a large number of devices.
- Devices may be grouped into Groups. This is interesting for charging, policing and multicast (toward a group).
- Devices should require ultra low power consumption.
- Time tolerant (generally), since data can be usually delayed. However, real-time transmission should also be considered for some applications.
- Time controlled as the devices may send and receive data only at certain periods of time.
- Security of exchanges between the devices and the server.

2.2 MTC applications taxonomy

It is worth to start by making things clear. MTC applications as they are thought in the context of sensors and servers (the typical cases we introduced in the previous paragraph), should be classified to make their study easier. We propose the classification of MTC applications into three main classes. Note that this is our classification we propose in the context of MTC. It comes from the data reporting way in Wireless Sensor Networks [8]. The criterion upon which we classify them is the traffic (signaling and data) and its origin. Note that a fourth class combining the three previous ones can exist.

1. **Event Driven applications:** The MTC devices send messages when an event occurs in the sensed area. For instance, messages are send when an intrusion is detected in a common application of security. Generally, this is considered to have more priority than the rest of the applications. However, the amount of data they send is small since the message to signal the event is generally short.

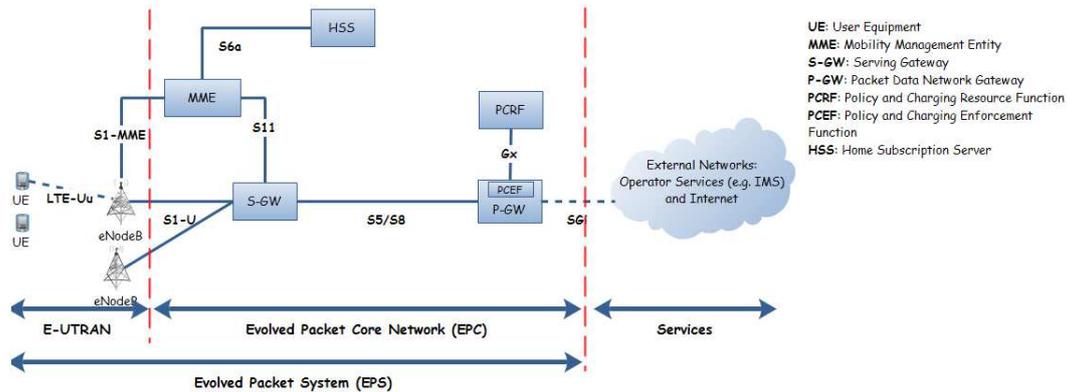


Figure 2: The LTE Global architecture

2. **Query Driven applications:** This is a common interactive application. Generally, the client sends a query to the server (see the temperature in the office). The server then sends the query to the sensor responsible to monitor the area concerned by the query (The sensor inside the office). The answer is then given by the MTC-Device. Note that the actuators are query driven applications since they behave in the same manner, i.e: they act according to the orders coming from the server.
3. **Time Driven applications:** The MTC-Devices send their messages (reports, sensed values...) to the server at regular rates (each hour for instance). The server is generally responsible for the rest of the treatment concerning the data.

2.3 Overview of the Long Term Evolution (LTE) networks

The LTE networks are the next generation of the deployed 3GPP networks. Their main target is a simplified architecture in comparison to its predecessors (UMTS) and the rates of 50 Mbps in uplink and 100 Mbps in downlink. They are under heavy standardization efforts led by 3GPP.

The global architecture of an LTE network is illustrated in Figure 2. It is composed of two main parts: the E-UTRAN (Evolved UMTS Terrestrial Radio Access Network) and a EPC (Evolved Packet Core Network). Furthermore, LTE networks are all-IP networks. This means that all the traffic that is transported in the network is IP packets. Moreover, traffic is divided into signaling and data. Signaling is the messages a UE (User Equipment) sends to the network and data is the useful messages that the UE send. For instance, in a voice call, the signaling is the messages sent by the UE to connect to the network and ask for the call, while, the data is the voice call when connexion is established.

1. **E-UTRAN part:** It is composed of the UEs and the eNodeB. eNodeB are the Base Stations that integrate functionalities like radio management.
2. **EPC part:** It is the Core Network part in the LTE network. It is responsible for the overall control of the UE and the establishment of bearers between the UE and the PDN Gateways (For simplicity, a bearer is a packet flow or tunnel between a UE and the P-GW. It is IP-only network. It comprises five logical nodes :

- Packet Data Network Gateway (P-GW): It insures the connexion with the IP Network of the operator. It is responsible for allocating IP addresses for UEs. It, also, filters downlink packets into bearers of different QoS and destinations (UEs). Furthermore, it is the anchor point for internetworking with non-3GPP networks such as CDMA2000 and WiMAX.
- Serving Gateway (S-GW): It serves as an anchor for data bearers when UE moves from eNodeB to another (handover). It also holds data of downlink bearers when UE is in idle state (turned off to save power, to release the radio bandwidth it uses for others...) and while the MME re-establishes the bearer with the UE. Similarly with P-GW, it serves also as an anchor, but this time, for internetworking with 3GPP networks (UMTS, GPRS...).
- Mobility Management Entity (MME): It is a key element in the LTE architecture as it is responsible for processing signaling between UE and the Core Network. It assumes functions related to both bearer management (establishment, maintenance and release) and functions related to attachment and connection management (establishment of the connection and security between the network and UE, authentication). It helps, as well, to reduce the overhead in the radio network by holding informations about UEs, which ensures a continuity when the UEs are in idle sates. These functions are handled by the Session Management Layer in the Non-Access Stratum (NAS) protocol [9]
- The Policy Control and Charging Rules Function (PCRF): It is responsible of the policy control decision making and the management of the flow based charging functionalities in the Policy Control Enforcement Function (PCEF). The latter resides in the P-GW. It provides the QoS authorization (QoS Class Identifier (QCI) and bit rates) that decides how a certain data flow will be treated in the PCEF (by the P-GW) and ensures that this is in accordance with the user's subscription profile.
- The Home Subscriber Server (HSS): It is like a database of the users. It contains informations about the users subscriptions. It could be the QoS profile or any access restriction for roaming. It holds also the PDN to which user can connect as well as dynamic information such as MME to which the user is currently attached or registered.

2.4 MTC applications in LTE networks

Mobile Network Operators (MNO) have already a large coverage deployed infrastructure for transporting users data. Hence, they can play a more direct role by carrying the data exchanged between remote devices and servers for MTC applications [7]. The benefit is for both MTC applications and MNO [5]. In one direction, M2M applications over cellular networks will create much more revenue opportunities for MNO. In the other direction, M2M applications will take benefit of an already and widely deployed infrastructure with rich services avoiding a cost of deploying a dedicated one. This is benefit for M2M applications that require mobility and high data rates with easier installation, especially for short time deployments [10]. Furthermore, it will also open new perspectives for device manufacturers and application developers [11].

According to the classification of MTC applications we gave in section 2.2, we present the simplified traffic patterns (through sequence diagrams) in case of deployment in LTE as follows:

1. Event Driven applications

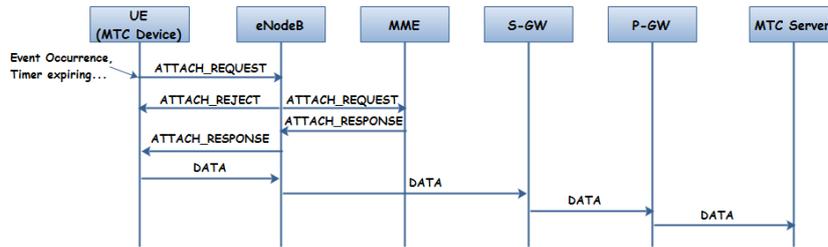


Figure 3: Sequence Diagram for signaling and data in case of Event Driven and Time Driven applications

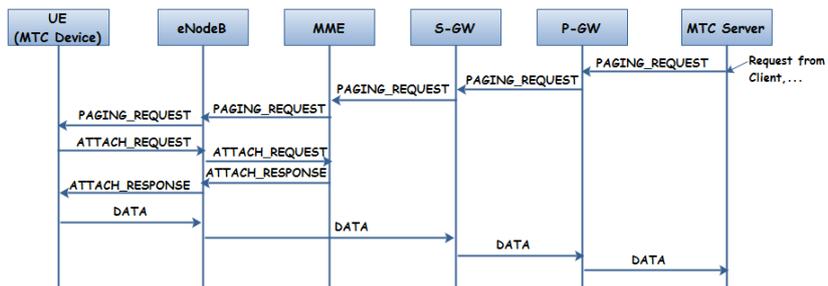


Figure 4: Sequence Diagram for signaling and data in case of Query Driven applications

The sequence diagram is illustrated in Figure 3. Each time an event occurs in the monitored area, the MTC devices connect to the network (Attach procedure) to send the alert messages to the server. During the rest of the time, the Devices remain in idle states and detached from the network. In idle state, a UE (MTC Device) is not connected to the network and has neither radio bearers nor S1 bearers.

2. Query Driven applications

In Query driven applications, the pattern is slightly different as illustrated in Figure 4. In fact, the requests for MTC devices interrogation come from the server. It is much more like a voice call received from another UE. The paging request is sent from the MTC server toward the MTC Device. The latter launches the attach procedure before sending data.

3. Time Driven applications

This is more or less the same as the Event Driven traffic pattern. In fact, the MTC Devices send data every hour for instance. During the rest of the time, they go into idle states. They have the same traffic pattern as Event Driven Applications. The sequence diagram is illustrated in Figure 3.

2.5 Overview of PID controllers

PID stands for Proportional, Derivative and Integrative controller. It is the most commonly used feedback controller in control theory. It is used to “correct” the behavior of the controlled process, mainly an industrial process. It has a simple principle: it computes an error as being the difference between the measured process variable and a desired setpoint we call the *reference*.

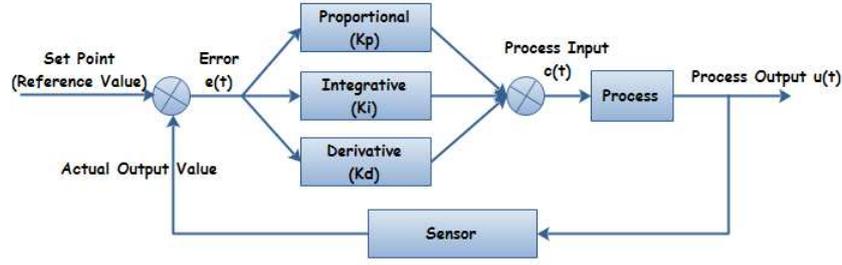


Figure 5: PID controller, general diagram

Then it attempts to reduce the error by adjusting the controlled process inputs. The general diagram of a PID controller in a closed loop is illustrated in Figure 5

The general equation of the PID controller is as follows (Note that other forms exist):

$$c(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (1)$$

where $e(t)$ is the instantaneous error, $c(t)$ the input of the controlled process, K_p , K_i and K_d are the Proportional, Integrative and Derivative gains respectively.

The three actions of a PID controller are:

- Proportional: in terms of time, it represents the present error. It is pondered by the *Proportional gain* we note K_p . It provides an instantaneous response to the error.
- Integrative: in terms of time, it represents the past cumulated errors (the sum of the instantaneous error over time) and gives the accumulated offset that should have been corrected previously. It is pondered by the *Integrative gain* we note K_i . It is used to accelerate the reach of the reference and to reduce the error in the steady state.
- Derivative: in terms of time, it represents the future predicted error. It is pondered by the *Derivative gain* we note K_d . It is used to reduce the overshoot produced by the integral component. However, it slows the transient state response of the controller.

In the last few years, the usage has reached the computing science. In fact, many approaches based on control theory have been introduced. Therefore, designing discrete controllers to adapt to the discrete nature of informatics is needed. Hence, a study of discrete PID controllers has been carried out. One of its results is the general discretized equation given by :

$$c(k) = c(k-1) + K_p(e(k) - e(k-1)) + K_i T_s e(k) + \frac{K_d}{T_s}(e(k) - 2e(k-1) + e(k-2)) \quad (2)$$

where :

- k is the k^{th} sampling instant
- $e(k)$ is the error at the k^{th} sampling instant

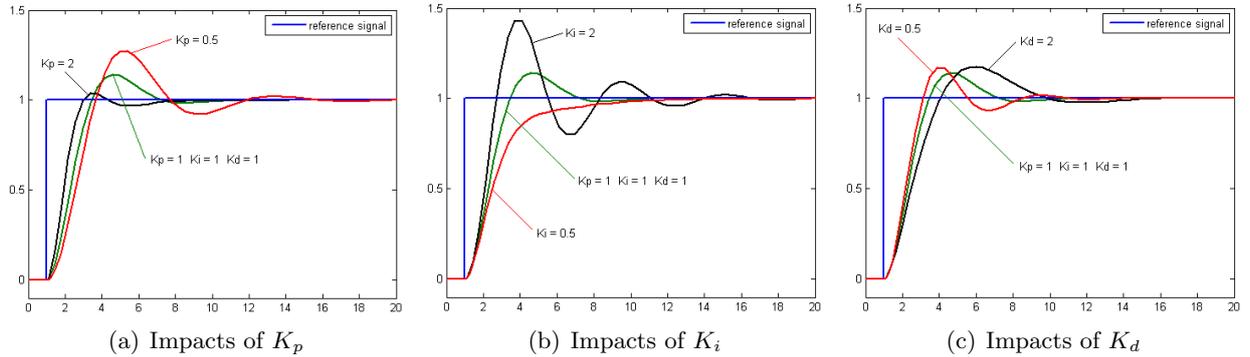


Figure 6: The impact of the gains (K_p , K_i and K_d) on the output value [1]

- T_s is the sampling period

It is obtained by mean of z -transformation of the continuous equation in 1.

The problem in PID controllers is the parameters' setting. These parameters are the different gains K_p , K_i and K_d and the sampling period T_s . The latter is generally chosen according to the nature of the process to control. However, the gains define the way the controller behaves and impacts the controlled process. For instance, high values of K_i lead to short time to reach the reference value but introduces oscillations around the reference in the steady state. Some of the effects of the parameters are illustrated in Figure 6.

Therefore, some methods are used. However, no optimal method exists. For instance, static methods have been proposed that use the pole placement if the controlled system transfer function is known. The most practically used are those methods based on empirical tuning of the parameters: the Proportional gain is set to a value so as to have an output value that approaches the reference even if the oscillations are big. Then, the oscillations are reduced by tuning the Integrative and then the Derivative gains.

Other methods for using more powerful and dynamic techniques have been proposed. In fact, methods based on neural networks are proposed in [12] and [13]. Approaches based on fuzzy logic have been proposed in [14] and [15], and others using a mixture of both fuzzy logic and neural networks in [16].

PID controllers and control theory in general have been widely used in computer science, in distributed systems [17] and to control the behavior of web servers [18], [19]. They have been also used in congestion control. In fact, [20], [21] and [22] proposed AQM based on PID controllers. The parameter setting of the controller is done using fuzzy logic [20], genetic algorithms [22] and genetic and neural networks [21]. Note that other works on the same topic of control theory applied congestion control exist. The list is not exhaustive and other examples are summarized in [23].

In the context of our work, The PID controller is used to adjust the behavior of an AQM. The reference value is given by the value of the queue length we seek to have and the output is given by the instantaneous queue length as explained in more details in Section 3.

2.6 The congestion problem

MTC applications are different from H2H, M2H and H2M applications. In fact, MTC Devices have to communicate without any external intervention. Furthermore, they are characterized by the fact that they involve a huge number of cheap and low-power devices which generate small amounts of traffic. These properties make MTC applications different and challenging in terms of deployment and management.

Furthermore, congestion is the core problem of LTE and all networks today. *“It is not about speed but capacity”*. This means that manufacturers are developing devices with high data rates, boosted by enhancements introduced in the radio access technologies. In the context of M2M applications, the problem is posed differently. Indeed, the applications are mainly characterized by a huge amount of devices that communicate frequently or infrequently by small amounts of data. These devices, for a given application, are deployed in a small area (home automation for instance). Thus, they are on competition in the same nodes of the network (eNodeB, MME, S-GW, P-GW, HSS). Consequently, a lot of devices trying to connect at the same time (generally application dependent) is frequently happening in M2M applications. This leads to bursts in signaling and data. This can penalize the non M2M devices.

In the context of MTC applications, congestion is due mainly to the big number of MTC devices as well as the global synchronization. The latter means that the devices attach and send data at the same time. This rises the problem of congestion at different levels and part of the network. The general scheme of an MTC application in LTE networks as well as the congestion points are illustrated in Figure 7.

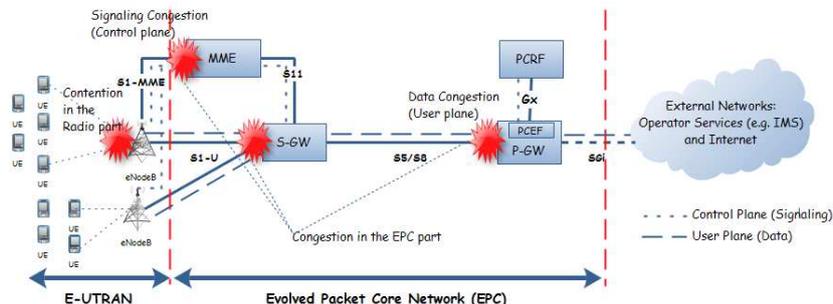


Figure 7: The congestion problem in LTE network in the context of MTC applications

Regarding the location, congestion can occur:

- in the radio network part: It appears commonly in eNodeBs where a lot of devices are connected to the same eNodeB and consequently use the same channels, leading to contention.
- in the Evolved Packet Core Network (EPC) part: It appears in the different nodes of the network, mainly in MME responsible of managing the attachment of devices, S-GW in charge of carrying the traffic and the P-GW (a lot of devices will send and receive traffic through the same P-GW). It can also concern the HSS responsible for managing the subscriptions, since each attachment of a device requires a procedure that involves the HSS, leading to congestion when a lot of devices have registered to the same HSS. Another problem is caused by the overhead of managing bearers between UE and the P-GW. A lot

of devices means a lot of bearers used for small periods of time, leading to an overhead for managing them by the MME mainly.

Regarding the nature of the traffic and the cause of congestion, we can mainly distinguish two classes: Congestion in the user plane and in the control plane. This separation is motivated by the fact that those planes (user and control) are separated in the EPC in LTE (Figure 7).

- Data traffic congestion (Data plane): It is caused by the amount of data sent and received by devices. In the context of MTC, this happens rarely since devices send and receive small amounts of data. But it may frequently happen that a lot of devices send their data at the same time leading to a congestion mainly in the EPC part. Even if the amount of data per MTC device is small, the sum of many of them can lead to congestion.
- Signaling congestion (Control plane): It appears in all the architecture. It is mainly due to the fact that the devices continuously generate signaling traffic to attach to the network, when triggering from the server, send data, send alerts, bearer management... Note that even if the devices send small amounts of data, they generate normal amounts of signaling. This causes overhead and congestion in MME mainly, and in all the other nodes.

In the context of our work, we are more interested in the Core Network Part. This doesn't mean we ignore the radio part but what we propose is a complementary solution to a solution dealing only with the radio part. Moreover, our solution reduces implicitly the contention in the radio part.

2.7 Related works

MTC applications in LTE networks is a new topic. Therefore, studies have started few time ago and the efforts toward standardization are also carried out. The problem of congestion in LTE networks has been cited by the ETSI [24] and 3GPP[25] whose efforts are conducted toward the standardization. However, some works dealing with congestion in LTE networks in general exist in literature. We classified them into three classes.

The first one is the contention management in the radio part. The goal of those approaches is a better usage of the radio resources while supposing that the core network is enough sized to handle all the traffic. They propose scheduling algorithms in the radio part and channel reservation [26, 27, 28] or even pricing policies [29]. However, as we explained in section 2.6, the congestion can appear and is mostly observed in the Core Network part. Hence, the radio part alone is not sufficient.

The second class proposes end-to-end congestion control. It is done by modifying the transport protocols. For instance, TCP-FIT, a modification in TCP for the context of LTE networks has been introduced in [30]. However, these techniques are not precise and fall to handle the problem of signaling congestion or do not treat it at all.

From an organizational point of view, the third class of techniques proposes application level manipulations to reduce the amount of data and signaling. For instance, the authors in [31] introduce a clustering technique based on bluetooth to gather data in a cluster head. The latter sends it to the network. Note that this Group based technique is biased with report to the Group

based concept introduced by 3GPP [2] since it is build in an Ad-Hoc manner, using Bluetooth technology for communications between the devices belonging to the same group. Moreover, is not interesting for operators since the modifications are done at the application level.

The most interesting of those works are the efforts of ETSI and 3GPP toward the standardization. Since release 8 [4, 32, 2, 33, 9, 34], specifications of the architecture evolution to enable deployment of MTC applications in LTE networks have been proposed. However, those specifications are still theoretical and no study on their efficiency (analytical, numerical...) have been published.

Not directly related to MTC applications (but in the context of cellular networks), some congestion control approaches have been proposed. For instance, [35] propose IP level congestion control mechanism. However, they are dedicated to separate the radio part from the CN part and not always adapted in the case of MTC.

Our work is considered as a complementary for those solutions that deal with congestion in the radio part only. In fact, the CN congestion control and admission control we use can be combined with a technique like [36] to make a full Congestion and Admission Control solution. Furthermore, we are interested in the IP level which is more practical for LTE networks since they are all-IP networks. Moreover, it is interesting from operators point of view. In fact, it doesn't introduce modifications at the application level but uses only the IP level. More details and comparisons are presented in the next section.

3 Our proposal: “*Congestion-Aware Admission Control*”

In this section we present our efforts to organize and propose a solution to the problem. We start by giving our classification of MTC applications and then the solution we propose to handle congestion.

3.1 Introduction

The solution propose combines both Congestion Control and Admission Control. The idea is quite simple. Each Core Network (CN) node (MME, S-GW, P-GW, HSS) will act as a probe to detect congestion and inform the eNodeB to reduce the traffic it sends to it. In a natural language, each CN node will signal congestion to the upstream nodes telling them: "Hey there, I want you to reduce the traffic you are sending me because I am congested". The strategy behind the idea is to reject the traffic as soon as possible, meaning that this is done in the radio part. In fact, the messages of reducing traffic will move from the CN nodes to the radio part (eNodeB). The latter then rejects traffic (Admission Control) to help the CN nodes.

The congestion metric we choose is the queue length at the IP level. This is motivated by the fact that LTE networks are designed to be all-IP networks. This means that both signaling and data, for either video, voice or any data are transported on IP as illustrated in Figure 8 describing the protocol stack in LTE networks. Thus, performing a classification of the incoming packets according to the high type (signaling or data) as well as application type will help us to detect congestion in both signaling and data and for different application types.

The packets from the different applications and services (data, voice, MTC...) are marked

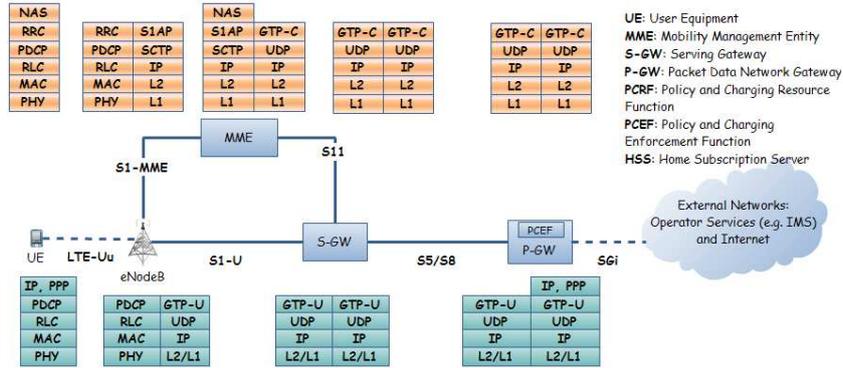


Figure 8: LTE Network Architecture

Type	Priority	Details
Non MTC	1	The traffic belonging to non MTC devices
Event Driven Application	11	This is the most prioritized traffic of MTC
..

Table 1: Priority and Traffic classification

according to the quality of services we provide for each of them. This is not new in LTE networks. In fact, a LTE network does this by marking the packets according to the QCI (QoS Class Identifier) and ARP (Allocation and Retention Priority) of their corresponding bearers in the CN (let's recall that a bearer is a tunnel with QoS). In our approach, we are to use the same technique by enhancing it to include MTC traffic with different levels of priority. For instance, the packets from an Event Driven application are marked in the same way and differently from Time Driven ones. In the intermediary nodes used to probe congestion, we implement Weighted Fair Queuing to schedule packets from different queues. Each queue represents a level of priority of the traffic. How the mapping could be done well as the different classes are illustrated in Table 1.

It is worth to note that in this work, we are exploring the possibility of using such a marking to implement our congestion control mechanism. However, the mapping for non MTC applications according to the QCI and ARP of the established bearers is specific to each operator and is left as future work. As well, the combination with the priorities and the type of applications we propose is also left as future work.

The main goals of our proposed solution are as follows:

- Avoiding non MTC traffic starvation when a lot of MTC devices try to connect at the same time.
- Ensuring a high utilization of the resources in the different nodes of the LTE network. This will increase the benefit of the MNO and the users satisfaction. In fact, one solution could be the blocking of a large amount of traffic but this results in an underutilization of the resources in the network. However, making a high utilization of resources in the CN part will lead to over-utilization of resources and hence overflows, lost packets in queues... as we see it in section 4.

- Ensuring robustness in the case of bursts. In fact, MTC applications have the tendency to send a lot of traffic at the same time and no traffic for long periods (See Section 4). They send traffic at the occurrence of an event which is detected by a lot of MTC devices or every synchronized periods. Therefore, our solution should be robust in case of bursts while insuring a high utilization of resources.
- Ensuring protection for non-MTC applications. This is targeted in the 3GPP standard (see [34], section 6.35). This holds, also, for the bearers of GBR (Guaranteed Bit Rate) where packet losses are not tolerated [37].

Practically, we use a queue monitoring technique based on a PID controller. The PID controller in a queue detects congestion and defines the Admission Control parameters. These parameters have to be applied (in the input of the system) to get the desired value of the queue length. In fact, the PID controller will take as an input the current queue length and a reference value. It produces the control variable which is a *Reject Factor*. The *Reject Factor* represents the sum of the amount of traffic to be rejected in comparison to the actual traffic to reach the reference value of the queue (the incoming traffic is the input of the system). The traffic to reject is the traffic that causes the system (the queue) to deviate from the reference value. This controller is implemented in every CN node (MME, S-GW and P-GW).

In congestion controller we used to see in literature, the focus is made on TCP mainly. This means that the traffic reducing is done by mean of the TCP congestion windows or any TCP-friendly mechanism for UDP. In the context of our work (LTE networks), the congestion reducing is done by mean of rejecting connexions in the radio part when congestion is experienced at any level of the network. This means that congestion is quantified in a probe system (the Core Network nodes). Then, decisions (congestion or not, the level of congestion) is sent back to inform about the state. The scheduling nodes (eNodeB) and the UE use the received information to reduce the number of connexions, the amount of traffic (eNodeB) and the signaling by rejecting traffic before reaching congested nodes. Moreover, UE are sent to idle states.

In literature, congestion controllers for networks in general have been proposed. For instance, Random Early Detection (RED) has been proposed by Sally Floyd et al. [38] and its variants Gentle RED [39] and Adaptive RED [40], and BLUE [41]. These congestion controllers focus on reducing the load while tolerating underutilization of resource (queue going down after congestion is experienced). The controller we propose is based on control theory. It insures a targeted utilization of the resources in the network while preventing losses. Some results and comparisons between the of PID congestion controllers and the previously cited ones are given in [20].

3.2 Congestion metrics

The congestion detection will be done in the different CN nodes (MME, S-GW and P-GW). In all the nodes, the congestion could be observed in the output links or while processing. The most noticeable is the fact that it is mostly observed in the output links. This means that the processing is generally done much faster than the capacity of forwarding to the next node. For instance, user data is forwarded from the S1 bearer to the S5 bearer (in the S-GW) after changing the encapsulation (at GTP-U level) and sent back to the IP layer.

Consequently, a congestion is materialized by a queue which is full or whose length is greater than a certain threshold. Furthermore, it is a measure of the resource utilization. In fact, a queue

which is always full and that loses packets (overflow) means that the resource (a link for instance) is over utilized, while, a queue always empty means that the resource is underutilized (if there are entities requesting the resource).

Let's note that the congestion controller we propose, as we will see it in more details in the paragraphs to come, could be used at different levels of the protocol stack where data is enqueued for processing or after being processed. In fact, the considered metric is queue length whatever the enqueued elements are.

One question could be why queue length and not the delay. In fact, the queue length is the most utilized in literature. For instance RED [38] and its variants Gentle RED [39] and Adaptive RED [40] and many others (BLUE [41]...) use the queue length as congestion metric. However, they use the average values of queue length (see [38] for examples). In our case, when combined with the PID controller, we use the evolution of the queue through the time.

In our case, we are interested in congestion in the Uplink direction. This means the traffic from the MTC Devices to the Core Network. In fact, most of the data is sent from the MTC Devices to the servers periodically, at the occurrence of an event or after a user query. Furthermore, The signaling is generated in the Uplink direction for attach requests. This doesn't mean that we ignore the downlink. The point is that congestion is mostly observed in the uplink direction in the context of MTC in contrary to non-MTC traffic (for instance, video streaming).

Furthermore, in the context of MTC applications, MTC Devices generally go to idle states (EMM_IDLE) (EMM holds for EPC Mobility Management). In fact, it is done in respect to the property of low energy consumption. When they need to send or receive data and/or signaling, they attach to the network. Therefore, the signaling congestion is quantified in the eNodeB and MME by the lengths of the queues of the corresponding packets. During the rest of the time, they go to idle state.

The signaling congestion is the most to be observed than the data congestion. In fact, since each data sent follows a signaling attach procedure, we are to observe a lot of signaling from MTC Devices than the Data. This is the opposite of the non MTC Devices where the amount of data is very important (voice call, video conference) in comparison to the amount of signaling.

3.3 Congestion-Aware Admission Control (CAAC)

As we presented earlier, and applied in the context of MTC applications, we define 3 kinds of applications classified in the rising order of priority: Time Driven, Query Driven and Event Driven. We handle congestion at the IP level. The first thing to do is to define a schema for packet marking. This means that according to the type of the application, the packets are marked by setting for instance the ToS (Type of Service). Then, according to the priority, the packets are scheduled in the core network part (MME, S-GW and P-GW). Our congestion controller will prevent congestion by computing a Reject Factor R_{rej} . R_{rej} depends on the packet class. The advantage with this approach is the fact that congestion is well defined (queue length) and accessible for measurement. The congestion notifications are sent back to the downstream nodes. This is for instance from the S-GW to the eNodeB. Practically we send a Reject Probability according to the congestion experienced in the node. The whole approach is illustrated in Figure 9.

To explain more, let's start from a simple model composed of n eNodeB. Those eNodeB are

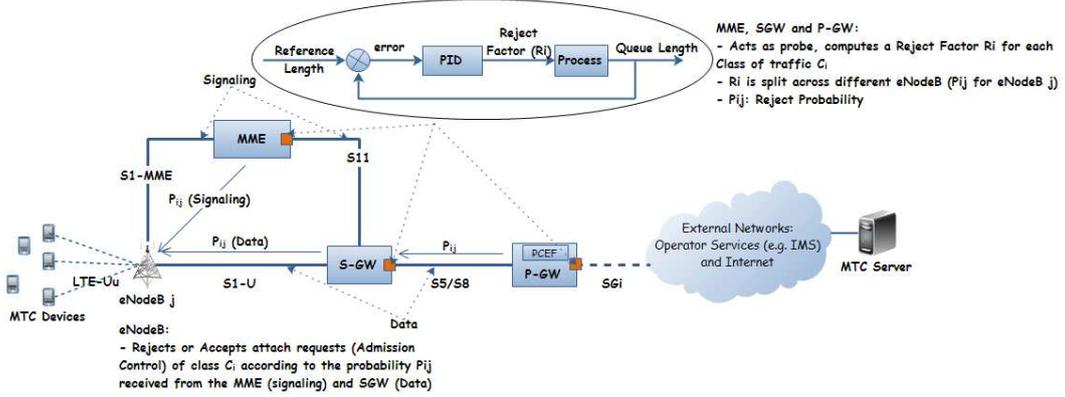


Figure 9: General Overview of the Solution

connected to the same MME, S-GW and P-GW. Both of the MME, S-GW and P-GW receive the traffic coming from the n eNodeB. It is worth to recall that all traffic is IP, both data and signaling. All what differentiates it is the marking we use (the ToS (Type of Service), port number,...). We summarize in the following what happens in each node. Lets take the example of the MME in case of signaling.

In the MME, a PID controller is implemented for each queue of class of traffic C_i . This means that for each class, the corresponding packets are enqueued in a separate queue. A queue monitoring technique using a PID controller is implemented for each class of traffic (the corresponding queue). It computes the Reject Factor we call R_i as follows :

$$R_i(k) = R_i(k-1) + K_p(e(k) - e(k-1)) + K_i T_s e(k) + \frac{K_d}{T_s}(e(k) - 2e(k-1) + e(k-2)) \quad (3)$$

where : $Q(k)$: the queue length at the k^{th} sampling instant ($t = k * T_s$)

Q_{ref} : the reference queue length we are targeting

$$e(k) = Q(k) - Q_{ref}$$

T_s : Resampling period

K_p : Proportional Gain, K_i : Integrative Gain, K_d : Derivative Gain

$$R_i(k) \in] - \infty, +\infty[$$

However, a negative value of the Reject Factor have no significance in our case. Therefore, it is set to 0 if bellow as follows:

$$R_i(k) = \max(R_i(k-1) + K_p(e(k) - e(k-1)) + K_i T_s e(k) + \frac{K_d}{T_s}(e(k) - 2e(k-1) + e(k-2)), 0) \quad (4)$$

Then, according to the sources or what we called the downstream nodes (eNodeB in this case), a dedicated Reject Probability $P_{i,j}$ (for eNodeB j) is computed as follows:

where n is the number the downstream nodes
 $traffic_{i,k}$ the amount of traffic of class C_i received from node k .

As $P_{i,j}$ is a probability, it is set to 1 if above. This gives :

$$P_{i,j} = \min\left(\frac{\text{traffic}_{i,j}}{\sum_{k=0}^n \text{traffic}_{i,k}} \cdot R_i, 1\right)$$

Finally, $P_{i,j}$ depends on time since R_i does. it is given by:

$$P_{i,j}(l) = \min\left(\frac{\text{traffic}_{i,j}}{\sum_{k=0}^n \text{traffic}_{i,k}} \cdot R_i(l), 1\right)$$

By doing so, we ensure a fairness between nodes and non-starvation of certain nodes.

The computed probabilities $P_{i,j}$ are then sent back to the downstream nodes (eNodeB). The eNodeBs implement the Admission Control mechanism. They reject traffic after having the true information of the traffic in the Core Network part summarized in $P_{i,j}$. Hence, only $(1 - P_{i,j}) * 100\%$ of the incoming traffic (from UE) is accepted. The Admission Control is implemented in each eNodeB j as follows :

```

On receipt of a packet (attach request) of class  $i$  from a UE
if ( $\text{unif}(0, 1) < P_{i,j}$ )
then
    Reject the request (attach request)
else
    Forward the request to the MME
endif

```

The interpretation of the Reject Factor is the amount of packets (from the class) to be rejected to reach the reference value of the queue length. For instance, it represents the sum of the reject probabilities in the downstream nodes in the case where $\forall j \text{ in } \{1 .. n\} P_{i,j} \in]0, 1[$. In fact

$$\begin{aligned} \sum_{j=0}^n P_{i,j} &= \sum_{j=0}^n \frac{\text{traffic}_j}{\sum_{k=0}^n \text{traffic}_k} \cdot R_i \\ &= \frac{\sum_{j=0}^n \text{traffic}_j}{\sum_{k=0}^n \text{traffic}_k} \cdot R_i \\ &= R_i \end{aligned}$$

However, this is not correct in that case where $P_{i,j} = 0$ or $P_{i,j} = 1$ (because of the rounding we perform to get the probabilities $P_{i,j}$)

4 Performance evaluation

4.1 Simulation model

We use ns-3 [3] to implement our solution and to carry out the simulations. The results are plotted using gnuplot. The choice of ns-3 was made after we used Opnet [42] for one month.

Opnet seemed to be not well adapted. In fact, it is designed to simulate as much as possible all the protocols of LTE. However, through our work, simulations take a long time before failing. It doesn't succeed to handle all the load we impose.

Therefore, we focused on ns-3, a well done simulator. In fact, ns-3 is designed to be a simulator for research instead of business matters. It is open source, done by research for research. However, it is a new simulator and doesn't have any dependency with its predecessor ns-2. It is developed in C++ and offers all the opportunities of Object Oriented Programming paradigm.

Under ns-3, we implemented the signaling and data messages as being application level messages. The congestion is quantified using the Queue at the output of the IP layer. It represents outgoing queue from the IP layer and the network device queue (Link Layer). This is an abstraction of the congestion we can have at any level of the architecture. For instance, we can replace it by a NAS (Non Access Stratum) level queue to handle the congestion at the NAS level (We consider the incoming queue of the NAS Server in the MME).

We developed a *PIDQueue* as being a module that has a Drop Tail Queue augmented by a PID controller. It is an extension (inherits) of the *Queue* class in ns3.

4.2 Test scenarios

We test in the following scenarios the case of Event Driven and Time Driven Applications. The sequence diagram of the generated traffic is illustrated in Figure 3.

In an Event Driven Application, the nodes likely to send the traffic at the same time are the nodes located in the same cell or in neighboring cells. In fact, when a fire is detected for instance in a region, the nodes that detect it are likely to be in the same area. This means that they are covered by the same eNodeB. Here we can, also, consider the case of VPN where an eNodeB plays the role of several logical eNodeB, so the nodes that communicate to signal the events could be from different cells.

Note that only one data packet is sent each time an attach procedure succeeds. It is a choice to simplify the simulations since we are more interested in signaling.

First, let's present the generic topology we use for simulations. It is illustrated in Figure 10.

At this level of our work, we are not interested in the radio part of the network. Thus, we assume that the eNodeB can handle the amount of traffic the UE generate. The core of our work deals with the congestion in the Core Network part as we explained before. The eNodeB are used to implement the Admission Control.

Consequently, the links between eNodeB and the MME and the S-GW are over-sized. We consider 10 Gbps point to point links. However, the links that are likely to represent bottlenecks are set to low rates, for instance, 1500 Kbps for the link between the MME and the SGW (signaling congestion) and 3000 Kbps between the S-GW and the data and signaling server. The rest of the parameters of simulations, we consider data and signaling packets of the same size (700 bytes).

We tested our approach in four scenarios: in case of uniform traffic, in case of bursts, in case of uniform traffic with bursts and a random traffic. In fact, bursts are what characterize MTC applications since devices are likely to send data at synchronized periods (timer fired or event occurred).

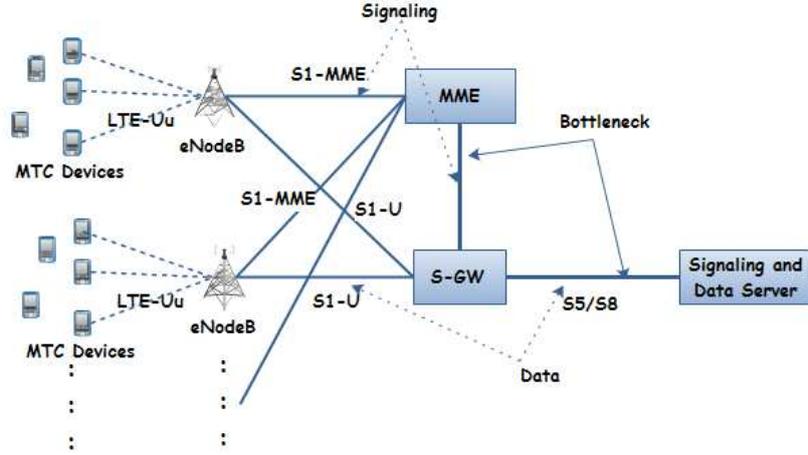


Figure 10: LTE Network Testing Topology

Not that in all the scenarios to come, we consider the UE as begin MTC Devices that generate traffic corresponding to applications of the same class. Furthermore, only traffic of this class is simulated. This is realistic since we can consider the case where VPN are used. For instance, we consider that each class of traffic is sent on a separate VPN.

4.2.1 Bursty traffic

Test configuration In this first scenario, we consider bursty traffic. The pattern of the traffic generated by the MTC Devices is illustrated in Figure 11.

The other parameters of the simulation are summarized in Table 2

Parameter	Value	Details
nMME	1	the number of MME
nSGW	1	the number of MTC Devices per cell
neNB	10	the number of eNodeB connected to each MME
nUEperCell	150	number of MTC Devices per cell
K_p	0.032	the PID Queue (MME) Proportional Gain
K_i	0.00051	the PID Queue (MME) Integrative Gain
K_d	0.000312	the PID Queue (MME) Derivative Gain
T_s	0.01 s	the PID Queue (MME) Resampling period
<i>Buffer size</i>	100	the buffer size (maximum queue length) (packets)
<i>Reference</i>	50	the PID Queue (MME) Reference length (packets)

Table 2: Simulation parameters in case of Bursty traffic

Results The evolution of the Reject Factor (R_i) in the MME and the Reject Probability ($P_{i,j}$) in each eNodeB is illustrated in Figure 12. We note that the Reject Factor responds to the traffic received. Furthermore, the eNodeB forward approximately the same amount of traffic. Hence,

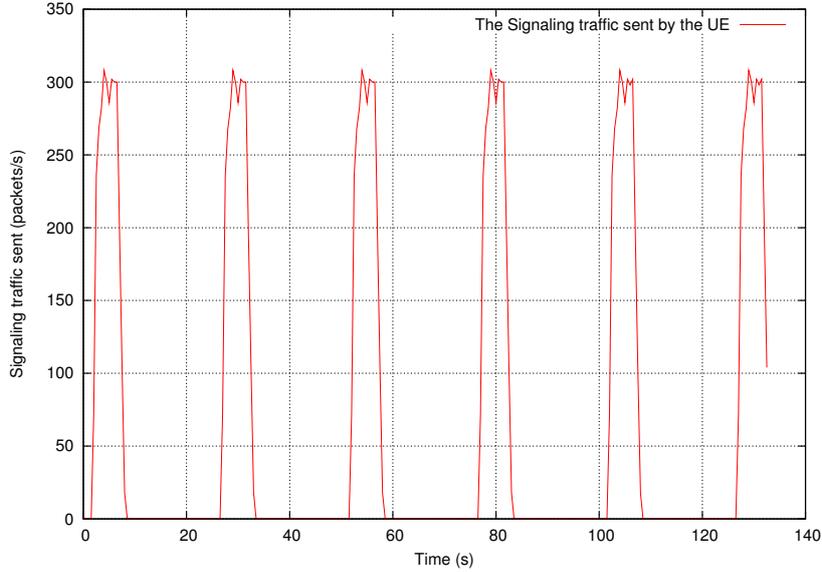


Figure 11: The Signaling traffic generated by the MTC Devices in case of bursty traffic

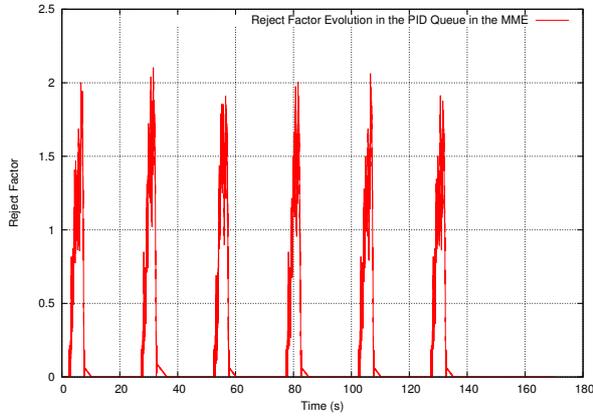
the Reject Probabilities are similar and represent around $\frac{1}{10}R_i$. This is what we expect since the eNodeB forward the same amount of traffic.

The evolution of the Reject Factor depending on signaling traffic received and queue length in the MME is given in Figure 13. What we note is the fact that the Reject Factor adapts to the amount of traffic received (that impacts naturally the queue length). We note also that the queue length is maintained around the reference by the PID controller and the eNodeB that rejects traffic. Furthermore, the bursts are well handled since neither overflow nor underutilization are observed.

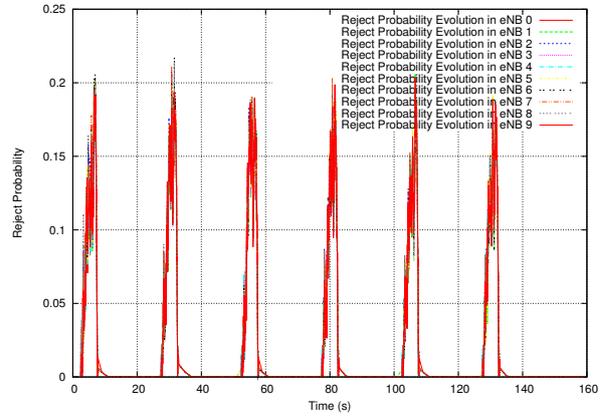
In fact, our approach handles well the congestion. The comparison with a case where we use drop tail queue are illustrated in Figure 14. We note that when using the PID queue in the MME and the Admission Control at the eNodeBs level, the queue length is maintained around the reference (50 packets) and no packets are dropped (lost), whereas, with a drop tail queue, bursts affect the queue length (over-utilization) and packets are dropped (lost) (Figure 14, (a) and (b)). At some points, 49 packets/s are lost. Furthermore, the amount of signaling forwarded by the eNodeBs is reduced since the same amount of data is sent for both of them (Figure 14, (c) and (d)). This means that we succeed to reduce the amount of signaling while keeping the same goodput (data sent thanks to the signaling).

4.2.2 Uniform traffic

Test configuration In the second scenario, the attach requests are uniformly distributed over the time. This means that all the connexions are independent from each other. The signaling traffic generated by the UE is illustrated in Figure 15. This holds for applications independent from each other that generates traffic toward the MTC servers in such a way that no bursts are observed. This could be a real scenario if some synchronization is done at the application level. For instance, in a time driven application, nodes are scheduled in different intervals to prevent



(a) Reject Factor in the MME



(b) Reject Probability in the eNodeB

Figure 12: The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Bursty traffic

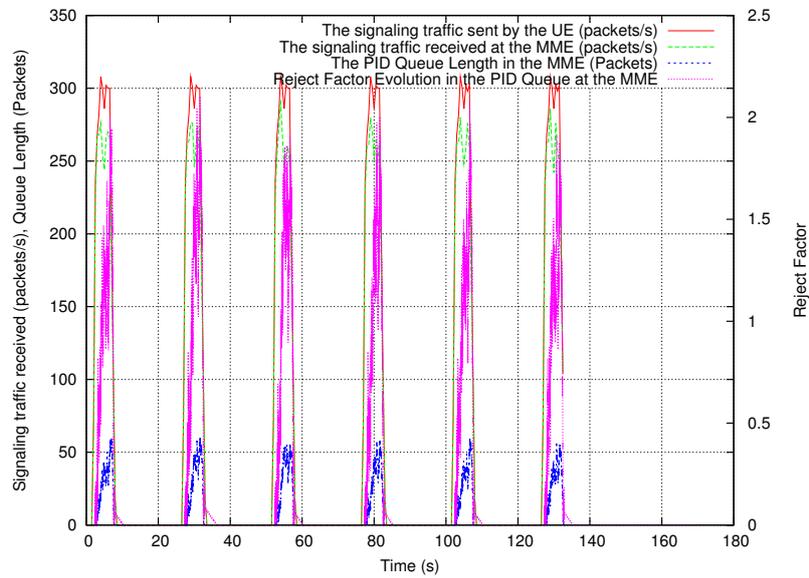
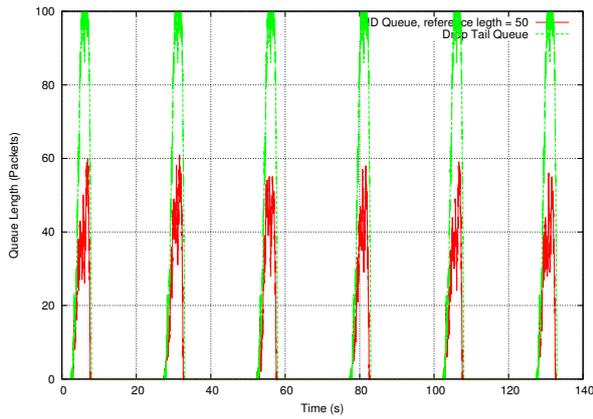
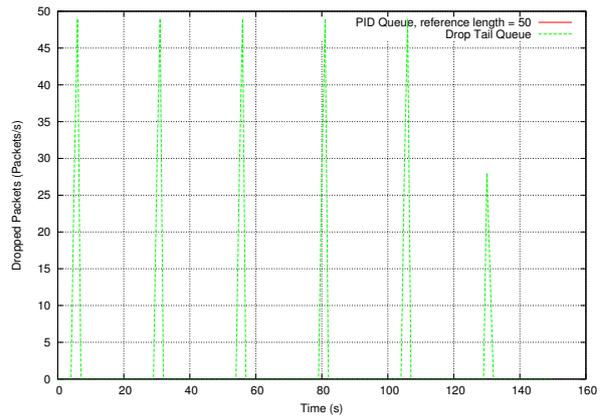


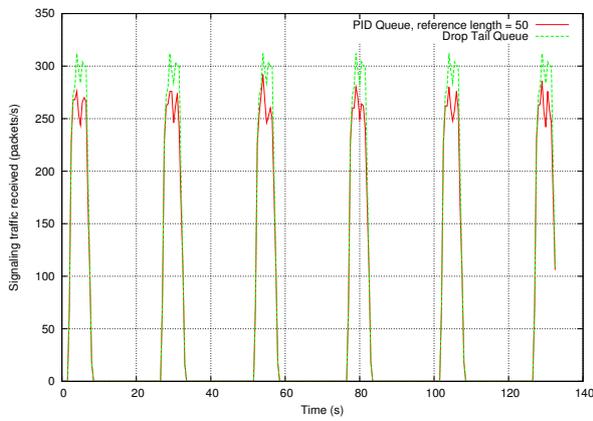
Figure 13: The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Bursty traffic



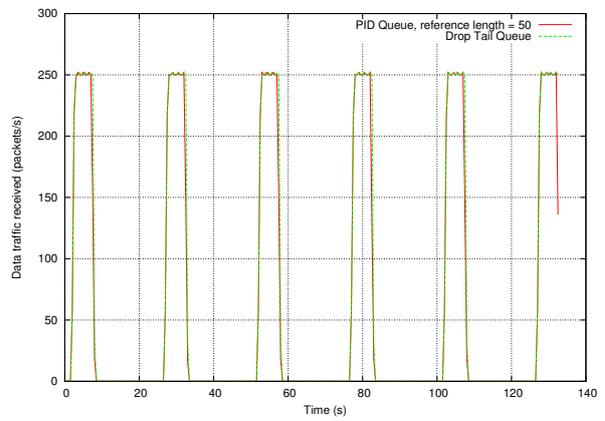
(a) Queue Length in the MME



(b) Dropped packets in the MME



(c) Signaling traffic received in the MME



(d) Data traffic received in the S-GW

Figure 14: Comparison between the case where a PID queue and drop tail queue are used in the MME in the case of Bursty traffic

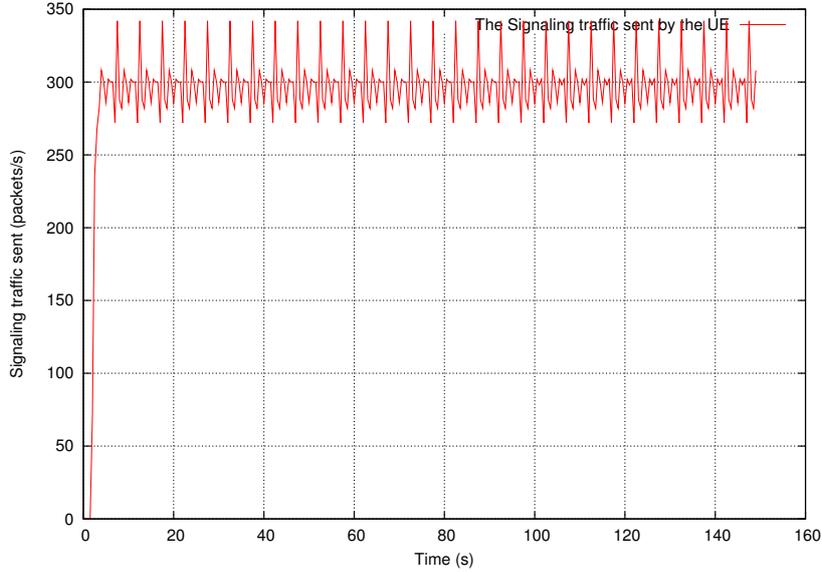


Figure 15: The Signaling traffic generated by the MTC Devices in case of Uniform traffic

bursts.

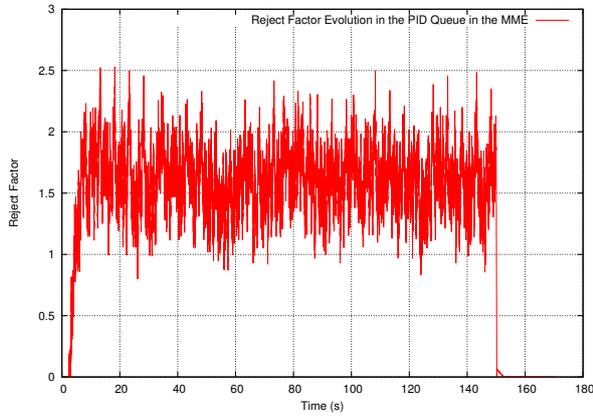
The other parameters of the simulation are summarized in Table 3.

Parameter	Value	Details
nMME	1	the number of MME
nSGW	1	the number of MTC Devices per cell
neNB	10	the number of eNodeB connected to each MME
nUEperCell	200	number of MTC Devices per cell
K_p	0.032	the PID Queue (MME) Proportional Gain
K_i	0.00051	the PID Queue (MME) Integrative Gain
K_d	0.000312	the PID Queue (MME) Derivative Gain
T_s	0.01 s	the PID Queue (MME) Resampling period
<i>Buffer size</i>	100	the buffer size (maximum queue length) (packets)
<i>Reference</i>	50	the PID Queue (MME) Reference length (packets)

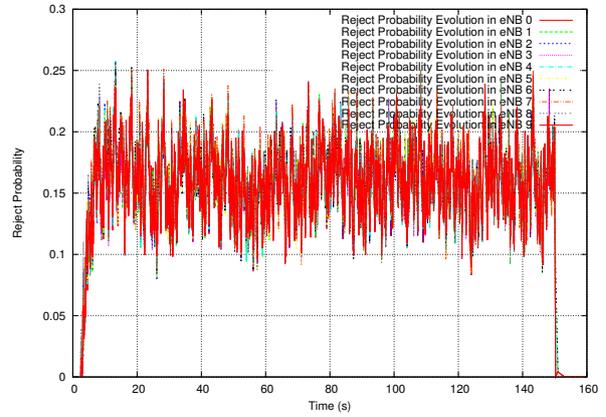
Table 3: Simulation parameters in case of Uniform traffic

Results The evolution of the Reject Factor (R_i) in the MME and the Reject Probability ($P_{i,j}$) in each eNodeB is illustrated in Figure 16. We note again that the Reject Probabilities are quite similar. Furthermore, they respond to the amount of traffic generated by the UE.

The evolution of the Reject Factor depending on signaling traffic received and queue length in the MME is given in figure 17. Again, we note that the Reject Factor adapts to the amount of traffic received (that impacts naturally the queue length). We note also that the queue length is maintained around the reference by the PID controller and the eNodeB that rejects traffic. Moreover, neither under-utilizations nor overflows are observed.



(a) Reject Factor in the MME



(b) Reject Probability in the eNodeB

Figure 16: The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Bursty traffic

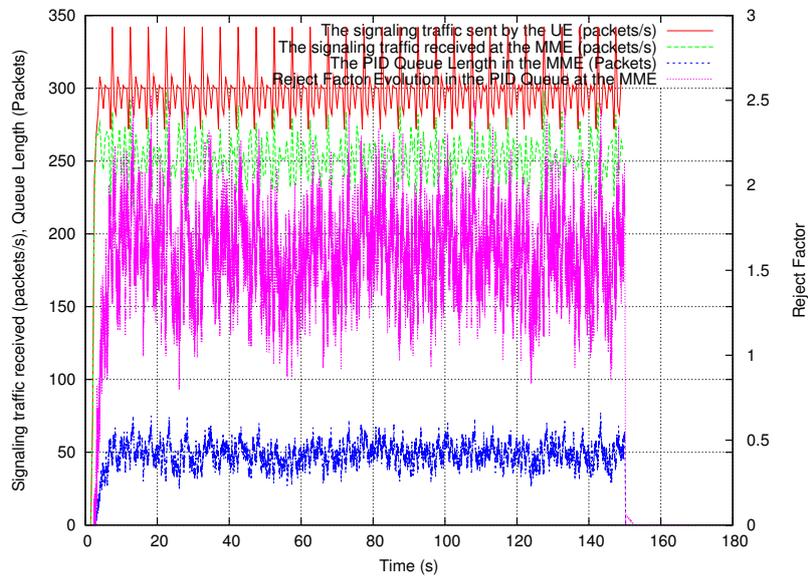
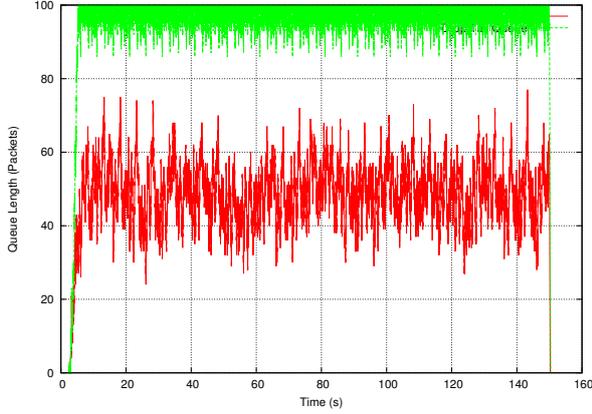
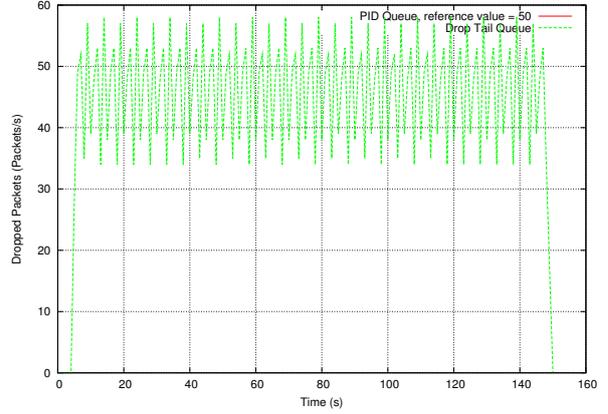


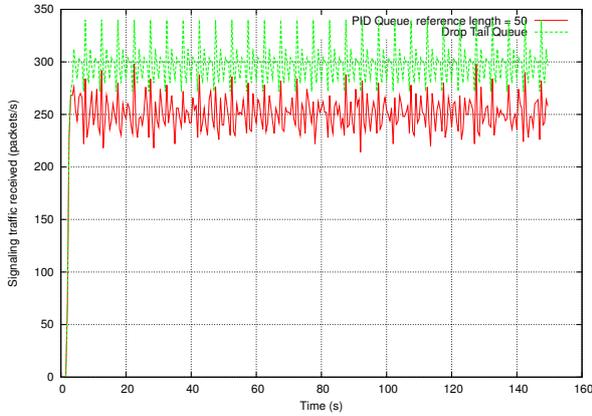
Figure 17: The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Uniform traffic



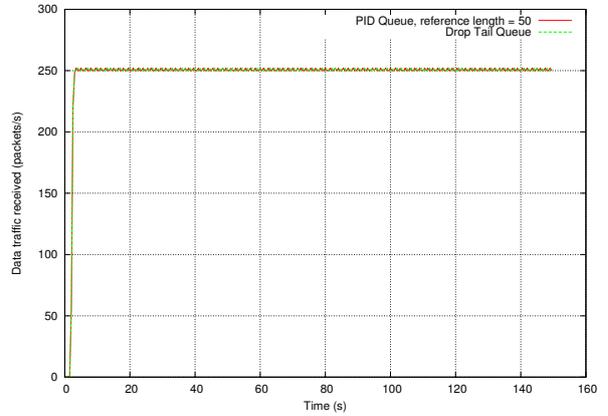
(a) Queue Length in the MME



(b) Dropped packets in the MME



(c) Signaling traffic received in the MME



(d) Data traffic received in the S-GW

Figure 18: Comparison between the case where a PID queue and drop tail queue are used in the MME in case of Uniform traffic

The comparison with a case where we use a drop tail queue are illustrated in Figure 18. We note that when using the PID queue in the MME and the admission control at the eNodeBs, the queue length is maintained around the reference (50 packets) and no packets are dropped (lost), whereas, with a drop tail queue, many packets are lost (45 packets/s in average) (Figure 18, (a) and (b)). Furthermore, the amount of signaling forwarded by the eNodeBs is reduced since the same amount of data is sent for both of them (Figure 18, (c) and (d)). This means that we succeed to reduce the amount of signaling while keeping the same goodput (data sent thanks to the signaling).

4.2.3 Uniform traffic with bursts

Test Configuration The third scenario is a combination of the two previous ones. The MTC Devices send traffic uniformly over the time. This is likely to be a Time Driven application. Sometimes, bursts are observed (an event occurring the monitored area). The signaling traffic generated by the MTC Devices is illustrated in Figure 19.

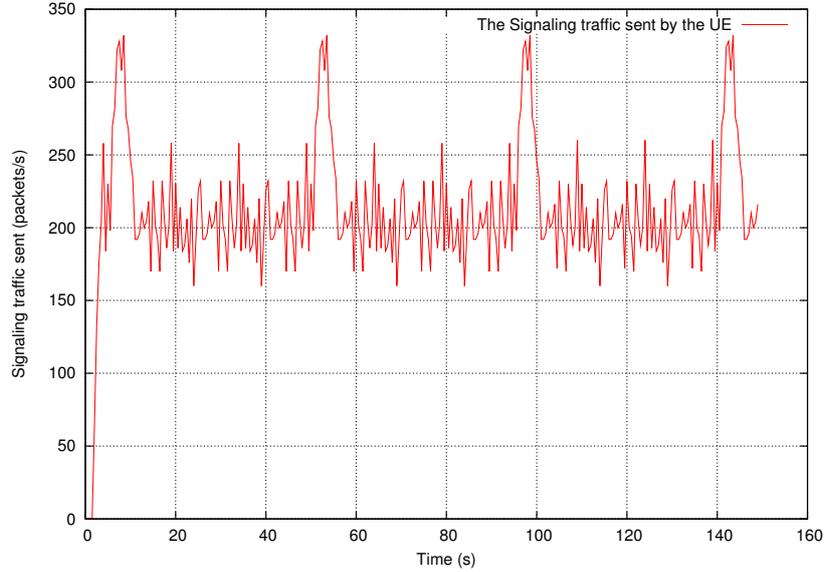


Figure 19: The Signaling traffic generated by the MTC Devices in case of Uniform traffic with bursts

The other parameters of the simulation are summarized in Table 4.

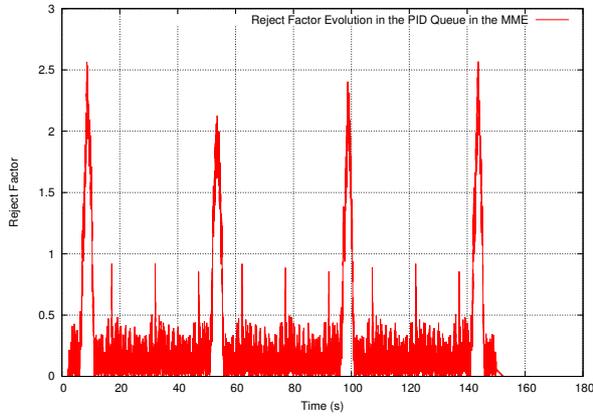
Parameter	Value	Details
nMME	1	the number of MME
nSGW	1	the number of MTC Devices per cell
neNB	10	the number of eNodeB connected to each MME
nUEperCell	350	number of MTC Devices per cell
K_p	0.032	the PID Queue (MME) Proportional Gain
K_i	0.00051	the PID Queue (MME) Integrative Gain
K_d	0.000312	the PID Queue (MME) Derivative Gain
T_s	0.01 s	the PID Queue (MME) Resampling period
<i>Buffer size</i>	100	the buffer size (maximum queue length) (packets)
<i>Reference</i>	50	the PID Queue (MME) Reference length (packets)

Table 4: Simulation parameters in case of Uniform traffic with bursts

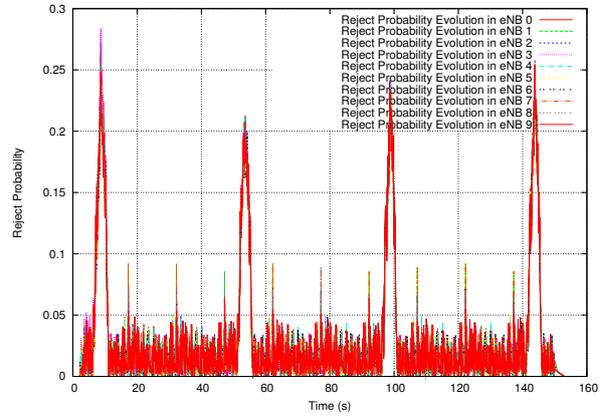
Results The evolution of the Reject Factor in the MME and the Reject Probability in each eNodeB is illustrated in Figure 20.

The evolution of the Reject Factor depending on signaling traffic received and queue length in the MME is given in figure 21. What we note is the fact that the Reject Factor adapts to the amount of traffic received (that impacts naturally the queue length). We note also that the queue length is maintained around the reference by the PID controller and the eNodeB that rejects traffic. Moreover, the bursts introduced are handled by the controller.

Furthermore, our approach handles well the congestion. The comparison with a case where we use drop tail queue are illustrated in Figure 22. We note that when using the PID queue in



(a) Reject Factor in the MME



(b) Reject Probability in the eNodeB

Figure 20: The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Uniform traffic with bursts

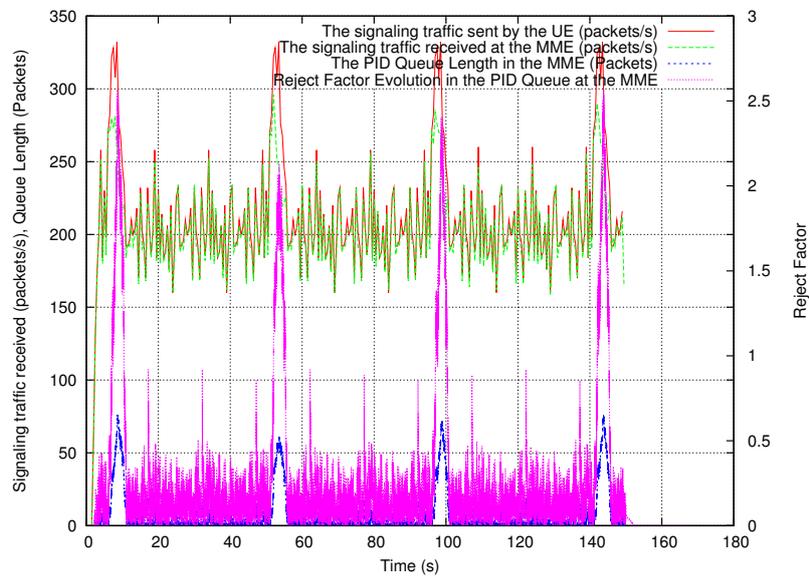
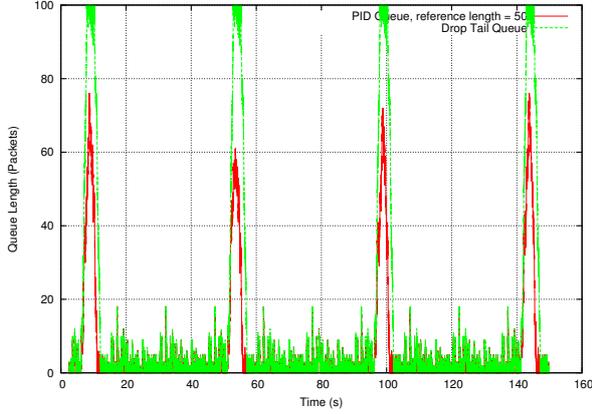
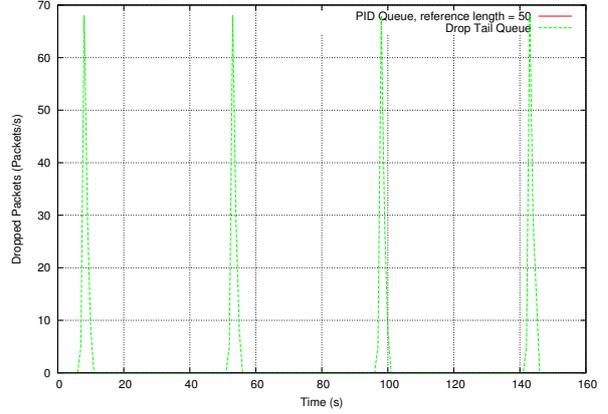


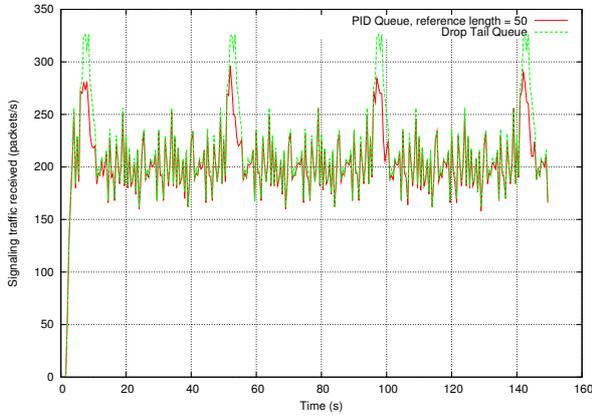
Figure 21: The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Uniform traffic with bursts



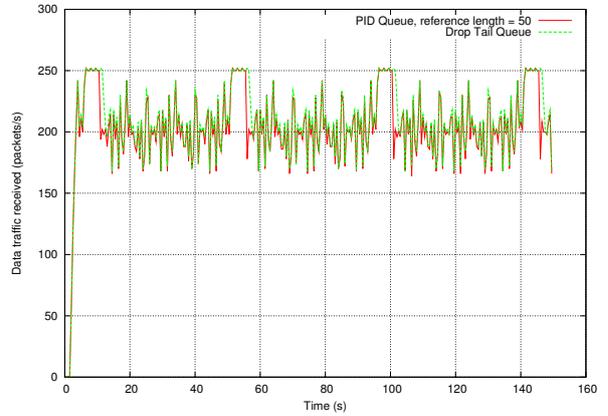
(a) Queue Length in the MME



(b) Dropped packets in the MME



(c) Signaling traffic received in the MME



(d) Data traffic received in the S-GW

Figure 22: Comparison between the case where a PID queue and drop tail queue are used in the MME in case of Uniform traffic with bursts

the MME and the admission control at the eNodeBs, the queue length is maintained around the reference (50 packets) and no packets are dropped (lost), whereas, with a drop tail queue, bursts affect the queue length and packets are dropped (at some points, 70 packets/s are lost) (Figure 22, (a) and (b)). Furthermore, the amount of signaling forwarded by the eNodeBs is reduced since the same amount of data is sent for both of them (Figure 22, (c) and (d)). This means, again, that we succeed to reduce the amount of signaling while keeping the same goodput (data sent thanks to the signaling).

4.2.4 Random traffic

Test Configuration In the fourth scenario, the MTC Devices send traffic randomly over the time. The signaling traffic generated by the MTC Devices is illustrated in Figure 23. We retrieve again the bursts and other patterns. This also a traffic likely to be observed if we deploy many applications independent from each other.

The other parameters of the simulation are summarized in Table 5.

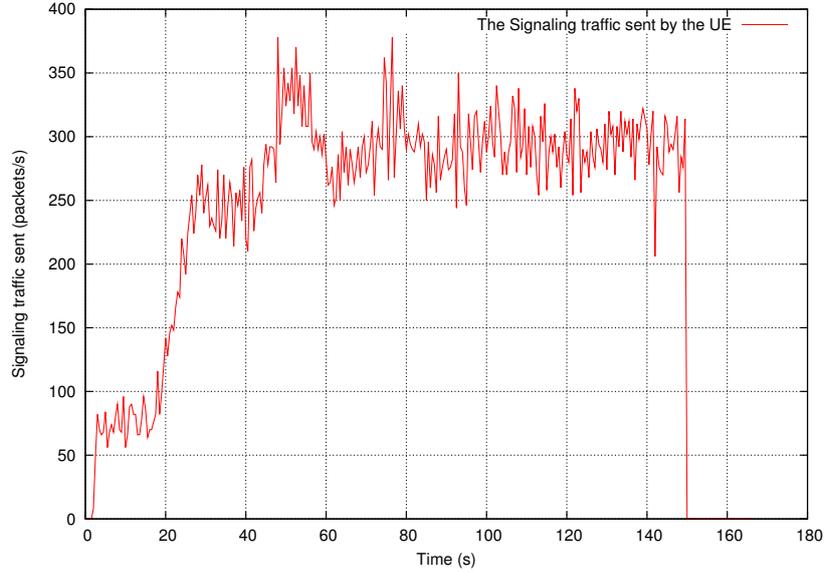


Figure 23: The Signaling traffic generated by the MTC Devices in case of Random traffic

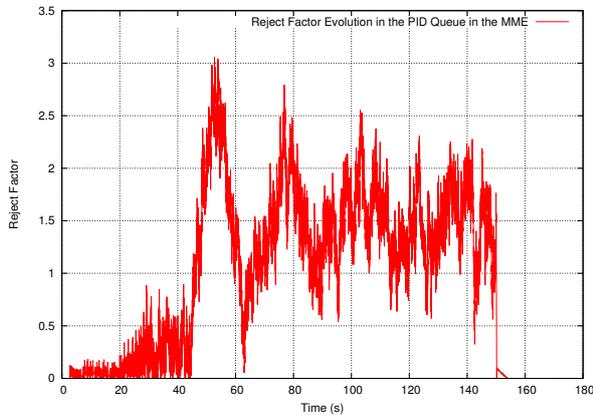
Parameter	Value	Details
nMME	1	the number of MME
nSGW	1	the number of MTC Devices per cell
neNB	10	the number of eNodeB connected to each MME
nUEperCell	350	number of MTC Devices per cell
K_p	0.032	the PID Queue (MME) Proportional Gain
K_i	0.00051	the PID Queue (MME) Integrative Gain
K_d	0.000312	the PID Queue (MME) Derivative Gain
T_s	0.01 s	the PID Queue (MME) Resampling period
<i>Buffer size</i>	100	the buffer size (maximum queue length) (packets)
<i>Reference</i>	50	the PID Queue (MME) Reference length (packets)

Table 5: Simulation parameters in case of Random

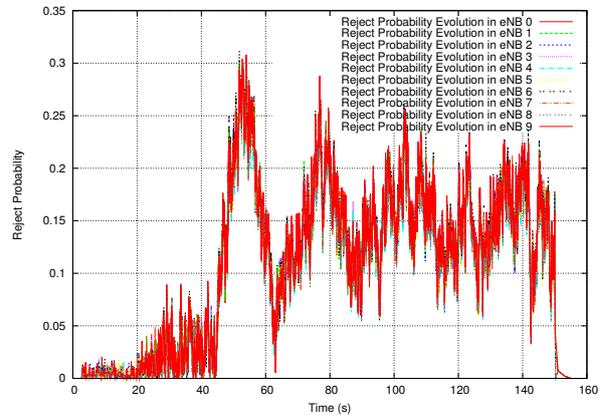
Results The evolution of the Reject Factor in the MME as well as the Reject Probability in each eNodeB is illustrated in Figure 24.

The evolution of the Reject Factor depending on signaling traffic received and queue length in the MME is given in figure 25. What we note is the fact that the Reject Factor adapts to the amount of traffic received (that impacts naturally the queue length). We note also that the queue length is maintained around the reference by the PID controller and the eNodeB that rejects traffic. The bursts noticed are handled by the controller.

Again, our approach handles well the congestion. The comparison with a case where we use drop tail queue are illustrated in Figure 26. We note that when using the PID queue in the MME and the admission control at the eNodeBs, the queue length is maintained around the reference (50 packets) and no packets are dropped (lost), whereas, with a drop tail queue, bursts affect the queue length and packets are dropped (at some points (between 48 and 53s), 90 packets/s are



(a) Reject Factor in the MME



(b) Reject Probability in the eNodeB

Figure 24: The Reject Factor evolution in the MME and the Reject Probabilities in the different eNodeBs in case of Uniform traffic with bursts

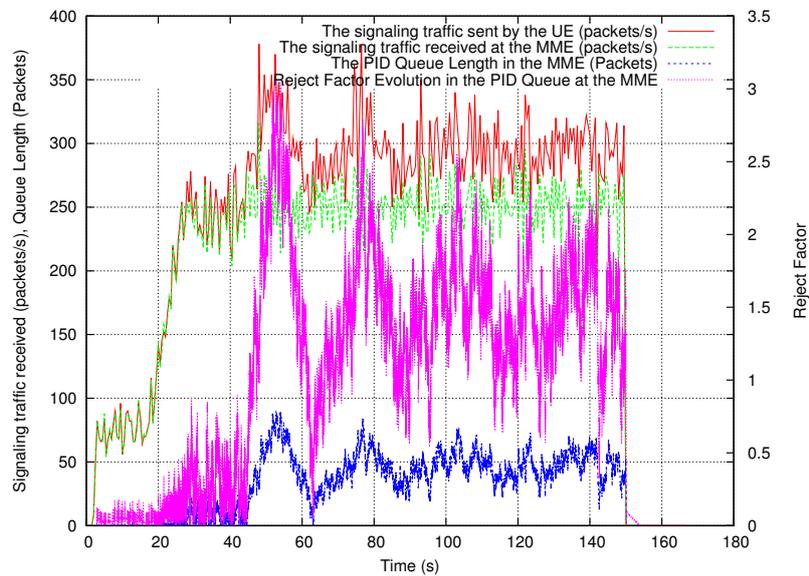
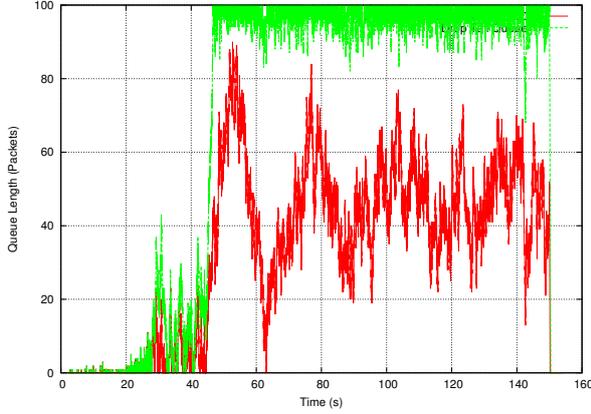
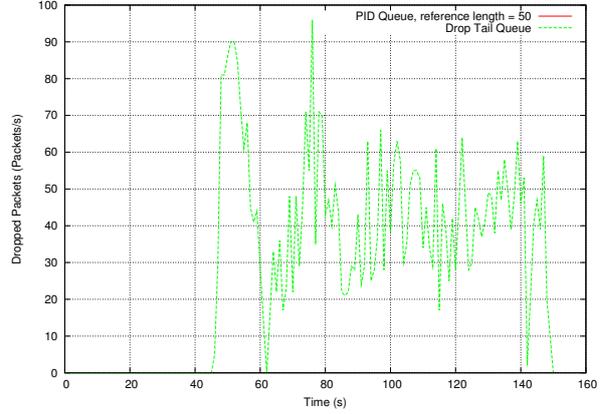


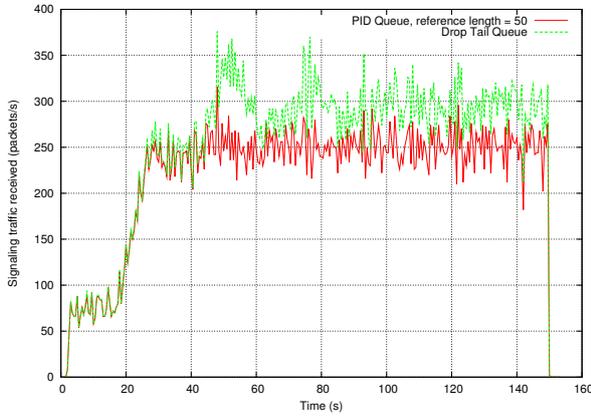
Figure 25: The Reject Factor evolution depending on the signaling traffic received and the queue length in the MME in case of Random traffic



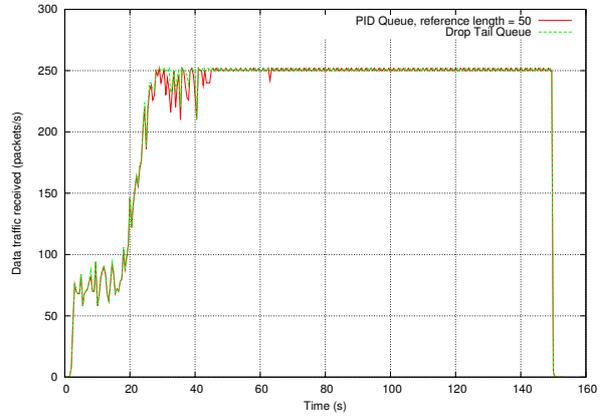
(a) Queue Length in the MME



(b) Dropped packets in the MME



(c) Signaling traffic received in the MME



(d) Data traffic received in the S-GW

Figure 26: Comparison between the case where a PID queue and drop tail queue are used in the MME in case of Random traffic

lost) (Figure 26, (a) and (b)). Furthermore, the amount of signaling forwarded by the eNodeBs is reduced since the same amount of data is sent for both of them (Figure 26, (c) and (d)). This means, again, that we succeed to reduce the amount of signaling while keeping the same goodput (data sent thanks to the signaling).

4.3 Discussion

We have seen through the different traffic patterns that we succeeded to reduce the amount of signaling while targeting a certain reference value of resources utilization in the Core Networks nodes. The PID controller in the MME generates a Reject Factor that dynamically changes according to the traffic generated by the UE and received by the MME and the queue length. This helps to maintain the queue length in the output of the MME at a desired reference value. Furthermore, our approach is robust against the changes in the traffic patterns event if with large sized networks. In fact, the charges we imposed to the network are likely to be those observed in real scenarios: 200-350 UE per cell and 10 cells that gives 2000-3500 UE covered by the same

MME and S-GW.

Moreover, all the requests are handled (attach request) and no requests are lost (dropped). This means that the network is not over-utilized. We also noticed that underutilization are not observed. In fact the resources utilization are maintained around the reference value during all the simulation.

Finally, We succeeded in all the scenarios to have the same goodput (data traffic sent) while sending a reduced amount of signaling.

5 Discussions on our proposed solution

Our proposed solutions presents many advantages. However some weakness and possible improvements are noticed.

5.1 Advantages

5.1.1 Impacts on the existing nodes and 3GPP standards

Our proposed solution reuses the existing interfaces between nodes of the EPC and e-UTRAN and doesn't introduce new protocols or interfaces. In fact, the exchanged messages carrying the reject probabilities according to the different levels we considered are transported in GTP-C messages (see Figure 8).

Furthermore, most of the concepts we reuse have been introduced in 3GPP specifications in [34]. What we did is to find a way to implement them and prove their efficiency. In fact, throttling mechanisms have been evoked with no more details. This means that nodes in the core network should be able to send throttle messages asking the downstream nodes to reduce the amount of traffic they forward. In our approach, this is implemented using the reject probability each core network node computes and sends.

Moreover, our solution introduces different levels of priority in contrary to what is proposed in 3GPP specifications in [34]. In fact, in 3GPP specifications, only low and normal priority levels are considered. Hence, attach requests sent by the equipment are rejected if their traffic is marked as being "Low Priority". What we propose is to use finer granularity and consider different levels. Thanks to the use of reject probabilities for each level, only a small part of the traffic is accepted or rejected. That traffic doesn't cause congestion while insuring a high utilization of the resources in the LTE network. However, the existing technique in [34] (reject all low priority traffic if congestion is experienced) can lead to under-utilizations since low priority traffic can represent lot of traffic.

5.1.2 Enhancement of several proposed approaches

Our solution could be extended to intermediary IP networks that could be crossed by the traffic. In fact, S.K. Kasera et al. [35] have introduced the problem of congestion in IP based CDMA networks. Their idea is based on congestion control at the IP level in the IP network linking the Base Stations with the Access Network Controller (ANC) in a UMTS network. Our solution

can encompass the latter if we extend the congestion control to intermediary IP networks that could exist between nodes belonging to the same MNO. This could be the case for an IP network between the eNodeB and the MME, the MME and the S-GW and so on. However, this is not possible if the IP network don't belong to the MNO since deep packet inspection could be expected (to handle the packet content, for instance if it is signaling, to see of which class of traffic it is,...). This depends on the implementation we choose.

5.1.3 Flexibility of the congestion metric

The congestion metric we consider is the queue length of the output links. However, it could be different from one node to another. This will be done according to its capacities. For instance, congestion in a eNodeB could be quantified by the measuring the available radio resources as mentioned in [36]. In fact, the most important thing we are interested in is the probability of rejecting a connexion, a packet or a data flow in general. For seek of simplicity, we considered only IP queues at different levels in the core network part as well as the eNodeBs. We can consider the queues of a server (application level) to quantify congestion and the approach keeps working in the same way. Up to now, our approach uses only the congestion in the Core Network part. It could be combined with a technique like [36] since they are independent parts in terms of congestion.

5.1.4 Protection against starvation

Our solution guarantees the QoS for non-MTC applications. In fact, the queuing policy used in the different nodes is tunable. This means that the specified reference value for a class ensures over-utilization and protection for the other classes. Furthermore, the fair queuing and the separation of queues of different classes guarantees that packets from non MTC are scheduled independently from MTC applications' packets if we want to do it.

5.1.5 Impacts on the different parts of the LTE networkss

- *On the Core Network Part*

Globally, our approach helps to keep a high utilization of the resources in the CN. In fact, the PID controller corrects the inputs to reach the desired reference value. A good choice of the PID parameters reduces the error (difference between the current value and the reference one) and maintains the desired utilization without losses. Also, it protects the network in case of bursts to avoid starvation.

- *On the Radio Part*

The congestion notification through probabilities a eNodeB receives helps to reject RRC connections. This saves Radio resources in case of congestion in the CN part. The Admission Control mechanism implemented at this level is more useful when all the classes (of MTC and Non-MTC) applications are used. In fact, MTC traffic is filtered and only RRC connection of devices that will not have a bad impact in the CN (MME mainly) are accepted.

- *On the UE part*

The devices receive reject responses with a probability that notifies about the severity of the congestion in the CN part. These devices schedule their next attach request according to the congestion factor and the back off time received from the eNodeB. In the 3GPP specification [34], the notion of randomization has been evoked. In our approach, the randomization is done at the eNodeB level according to the severity of the congestion.

As a consequence, the eNodeBs seek to find a *good* back off time that helps the CN to return into a non-congested state while keeping a good utilization of the resources. The higher the Reject Probability is, the longer the back off time is. Also, it helps the UE to save their energy. In fact, on one side, sending messages is energy consuming, and on the other side MTC applications require low energy consumption devices. The UE go to idle state and save their energy thanks and avoid sending vain attach requests when congestion is experienced in the CN.

5.1.6 Benefits for Mobile Network Operators

The solution we propose uses congestion control at the IP level. It doesn't impact how the applications should behave but corrects their behavior. This is very important since the applications (MTC applications) can belong to a third party. Therefore, the MNO cannot impose application level constraints on them. For instance, to solve the problem of global synchronization (MTC devices synchronized to send periodic reports at the same time), the MNO can impose to applications to do that (avoid sending the reports at the same time). However, this is not practical since such a synergy between application developers and MNO is not always assured. To overcome these problems, our solution don't use application level constraints. It is replaced by the packet marking.

Furthermore, it is a one shot solution. In fact, the traffic is treated as being the same for both signaling and data. This keeps things simple and reduces costs.

Finally, our solution is designed in such a way to allow cohabitation on different MNO on the same infrastructure. In fact, the same LTE network could shared between two or more operators. Each of them will have a certain amount of resources to use depending for instance on a billing system. The hint is to define different queues at the IP level and schedule the different queues proportionally to the MNO the traffic belongs to. This may rise problems of networks neutrality which is out of scope of this document. In fact, one operator could privilege traffic belonging to his subscribers while discarding the other users belonging to another MNO with whom the infrastructure is shared.

5.2 Weaknesses and possible improvements

Up to now, our solution addresses only congestion in the control plane (signaling). This is the first step toward a full solution. Furthermore, optimizations and deep study of the way we compute the Reject Probabilities is not addressed yet. For instance, if we take the probabilities sent by the MME and S-GW to the same eNodeB, in our solution, we choose the maximum of them to be the corresponding Reject Probability of the class. This behavior reduces congestion and prevents losses. However, it could be too aggressive. This may lead to an underutilization of the

resources in the CN part (These are our predictions). Moreover, we considered only congestion in the uplink. This is realistic but some applications may pose the problem in the downlink also. Finally, the PID parameters (the different gains) have been set manually. In fact, a tuning was performed to have the values we used on the simulations. In the next step, a dynamic technique to set them is expected. It will dynamically change those parameters (gains) to have optimal values corresponding to the traffic in the network. For instance, it could be based on fuzzy logic or genetic algorithms.

6 Conclusion

MTC applications involve communicating devices and servers without any human intervention. They use a network to transport data from devices to servers. The Cellular networks and the next generation of the deployed 3GPP (LTE) offer a ready to use and large coverage infrastructure. Therefore, deploying MTC applications in LTE networks presents benefits for both MNO and application developers. However, such a deployment rises problems due to the nature of LTE networks that are not designed to support MTC, which involve huge number of devices. One of the most important problems is congestion. Congestion is observed in all the network parts, both the radio and the core network.

In the presented work, we proposed a Congestion-Aware Admission Control that handles congestion, in LTE networks, in the context of MTC applications. Our proposal is based on queue monitoring which a PID controller to detect congestion in each Core Network node (MME, S-GW and P-GW) and Admission Control in the radio part to reduce traffic. The solution was evaluated using simulations carried out on ns-3. The traffic patterns and the test scenarios were thought to be generic and representative of common MTC applications. The obtained results show that our proposal is efficient. In fact, the amount of signaling is reduced and the utilization of resources is maintained around the reference (targeted) value while avoiding under-utilizations and overflows. Furthermore, our approach is robust for the traffic patterns we tested: bursts, uniform, uniform with bursts and random.

As future work, we are interested in using a technique to dynamically adapt the PID controller parameters. Those parameters are the different gains (proportional, integrative and derivative). This could be based on fuzzy logic to adapt the parameters to the load and the traffic pattern detected in the network.

References

- [1] Wikipedia web site : <http://en.wikipedia.org/>.
- [2] 3GPP website. www.3gpp.org.
- [3] ns3 web site : <http://www.nsnam.org/>.
- [4] 3gpp ts 22.368 v10.1.0 (2010-06): Service requirements for machine-type communications (mtc). 2010.
- [5] Mladen Sokele, Vlasta Hudek, and Alexandru-Ioan Mincu. Opportunities for implementation machine-to machine services via 3g mobile networks. In *Proceedings of the 7th International Conference on Telecommunications, 2003. ConTEL 2003.*, volume 1, pages 91–95. IEEE Computer Society, June 2003.
- [6] Numerex website, <http://www.numerex.com/>.
- [7] Bob EMMERSON. Perspective-m2m: the internet of 50 billion devices. European Editor of M2M Magazine, January 2010.
- [8] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6 – 28, dec. 2004.
- [9] 3gpp technical specification 24.301 : Non-access-stratum (nas) protocol for evolved packet system (eps); stage 3 (release 8). 12 2008.
- [10] Standardization of machine-type communications v0.0.5 (2010-10). October 2010.
- [11] Vania Goncalves and Philippe Dobbelaere. Business scenarios for machine-to-machine mobile applications. *Mobile Business / Global Mobility Roundtable, International Conference on*, 0:394–401, 2010.
- [12] Giulio D’Emilia, Antonio Marra, and Emanuela Natale. Use of neural networks for quick and accurate auto-tuning of pid controller. *Robot. Comput.-Integr. Manuf.*, 23:170–179, April 2007.
- [13] Gary M. Scott, Jude W. Shavlik, and W. Harmon Ray. Refining pid controllers using neural networks. *Neural Comput.*, 4:746–757, September 1992.
- [14] K.S. Brehm, T.; Rattan. Hybrid fuzzy logic pid controller. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, volume 3, pages 1682–1687, June 1994.
- [15] Pivonac P. Comparative analysis of fuzzy pi/pd/pid controller based on classical pid controller approach. In *Proceedings of the 2002 IEEE International Conference on*, pages 541–546, May 2002.
- [16] Ching-Hung Lee, Yi-Hsiung Lee, and Ching-Cheng Teng. A novel robust pid controller design by fuzzy neural network. *Asian Journal of Control*, 4(4):433–438, 2002.

- [17] Sherif Abdelwahed, Nagarajan Kandasamy, and Sandeep Neema. A control-based framework for self-managing distributed computing systems. In *Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems*, WOSS '04, pages 3–7, New York, NY, USA, 2004. ACM.
- [18] S. Parekh, N. Gandhi, J. Hellerstein, D. Tilbury, T. Jayram, and J. Bigus. Using control theory to achieve service level objectives in performance management. *Real-Time Syst.*, 23:127–141, July 2002.
- [19] Tarek F. Abdelzaher, Kang G. Shin, and Nina Bhatti. Performance guarantees for web server end-systems: A control-theoretical approach. *IEEE Trans. Parallel Distrib. Syst.*, 13:80–96, January 2002.
- [20] Yassine Hadjadj Aoul, Ahmed Mehaoua, and Charalabos Skianis. A fuzzy logic-based aqm for real-time traffic over internet. *Comput. Netw.*, 51:4617–4633, November 2007.
- [21] Xiao Laisheng, Wang Zhengxia, and Peng Xiaohong. Research on congestion control model and algorithm for high-speed network based on genetic neural network and intelligent pid. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1 –6, sept. 2009.
- [22] Wang zhiquan Lu Jinjun. New control strategy of network congestion based on optimization of ga-id parameters. *Journal of Nanjing University of Aeronautics And Astronautics*, 13(6):858–862, January 2008.
- [23] Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4:14:1–14:42, May 2009.
- [24] Etsi (european telecommunications standards institute) web site : <http://www.etsi.org/>.
- [25] 3gpp (3rd generation partnership project) web site : <http://www.3gpp.org/>.
- [26] Fei Yu and Victor Leung. Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks. *Comput. Netw.*, 38:577–589, April 2002.
- [27] Sunghyun Choi and Kang G. Shin. Predictive and adaptive bandwidth reservation for hand-offs in qos-sensitive cellular networks. *SIGCOMM Comput. Commun. Rev.*, 28:155–166, October 1998.
- [28] Guohong Cao. Integrating distributed channel allocation and adaptive handoff management for qos-sensitive cellular networks. *Wirel. Netw.*, 9:131–142, March 2003.
- [29] Jiongkuan Hou, Jie Yang, and Symeon Papavassiliou. Integration of pricing with call admission control to meet qos requirements in cellular networks. *IEEE Transactions on Parallel and Distributed Systems*, 13:898–910, 2002.
- [30] Jingyuan Wang, Jiangtao Wen, Jun Zhang, and Yuxing Han. A demonstration of a new tcp congestion control algorithm over lte and other challenging networks. *ACM S3 2010 Workshop, Chicago, Illinois*, September 2010.
- [31] Kwang-Ryul Jung, Aesoon Park, and Sungwon Lee. Machine-type-communication (mtc) device grouping algorithm for congestion avoidance of mtc oriented lte network. In Tai-hoon Kim, Adrian Stoica, and Ruay-Shiung Chang, editors, *Security-Enriched Urban Computing*

and *Smart Grid*, volume 78 of *Communications in Computer and Information Science*, pages 167–178. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-16444-6_22.

- [32] Etsi ts 102 689 v1.1.1 (2010-08): Machine-to-machine communications; m2m service requirements. August 2010.
- [33] 3gpp tr 36.306 v8.2.0 (2008-05) evolved universal terrestrial radio access network (e-utra); user equipment (ue) radio access capabilities. 05 2008.
- [34] Technical Specification Group Services and System Aspects. 3gpp tr 23.888 v0.5.0 (2010-07): System improvements for machine-type communications. (Release 10), July 2010.
- [35] S.K. Kasera, Ramachandran Ramjee, S. Thuel, and X. Wang. Congestion control policies for ip-based cdma radio access networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 712 – 722 vol.1, march-3 april 2003.
- [36] R. Kwan, R. Arnott, R. Trivisonno, and M. Kubota. On pre-emption and congestion control for lte systems. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1 –5, sept. 2010.
- [37] S. M. Chadchan and C. B. Akki. 3gpp lte/sae: An overview. *International Journal of Computer and Electrical Engineering*, 2(5):806–814, sept. 2010.
- [38] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *Networking, IEEE/ACM Transactions on*, 1(4):397 –413, aug 1993.
- [39] S Floyd. Red: Optimum functions for computing the drop probability? <http://www.icir.org/floyd/REDfunc.txt>.
- [40] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker. Adaptive red: An algorithm for increasing the robustness of red’s active queue management. Technical report, 2001.
- [41] Wu-chang Feng, Kang G. Shin, Dilip D. Kandlur, and Debanjan Saha. The blue active queue management algorithms. *IEEE/ACM Trans. Netw.*, 10:513–528, August 2002.
- [42] Opnet web site : <http://www.opnet.com/>.