E-Diophantine
An Admission Control Algorithm for WiMAX Networks

Xavier Pérez-Costa[†], Marco Mezzavilla[†], J. Roberto Boisson de Marca[⊕], Julio Aráuz[*]

[†] NEC Laboratories Europe, Network Research Division, Heidelberg, Germany
[⊕] Catholic University, Rio de Janeiro, Brazil
[*] School of Information and Telecommunication Systems, Ohio University, U.S.A

E-mail: perez@neclab.eu, mezzavilla@neclab.eu, r.demarca@comsoc.org, arauz@ohio.edu

Abstract—**Admission control algorithms must ensure that, when a new QoS resource reservation is accepted, reservations already present in the system continue having their QoS guarantees honored. In this paper we consider different approaches to compute the aggregated allocated capacity in WiMAX networks and, based on their limitations, propose the *E-Diophantine* solution. The mathematical foundations for the designed approach are provided along with the performance improvements to be expected, both in accuracy and computational terms, as compared to three alternatives of increasing complexity. Finally, the different solutions considered are evaluated with OPNET's WiMAX simulator in a realistic scenario.**

Keywords—**Wireless MAN, Wireless LAN, 802.16e, 802.11e HCCA, QoS, Admission Control.**

I. INTRODUCTION

WiMAX is a promising new broadband wireless technology currently being deployed worldwide [1]. Based on the OFDMA concept, it allows service providers to support a variety of non-real-time and real-time applications in diverse mobile and fixed environments. In WiMAX, applications can be assigned to any of several scheduling services that provide different QoS guarantees. For instance, services such as UGS (unsolicited grant service) provide hard QoS guarantees while others like rtPS and nrtPS (real-time and non real-time polling services) provide less strict QoS levels; finally, BE (best effort) offers no guarantee. Complying with the QoS requirements of granted service demands is mandatory for service providers which requires accurately estimating the system's capacity. Precise capacity estimation techniques allow the design of efficient admission control algorithms.

Current literature on admission control for WiMAX proposes a wide range of options achieving very different levels of accuracy as well as computational load. The authors in [2] propose a simple approach that is mainly based on the mean data rate requirements that an application specifies. Such approach requires few computational resources; however, neither does it take into consideration the time-varying nature of typical applications such as video or voice with activity detection nor the time period at which these resources are required. Thus, actual available resources might be unused. A similar solution is considered in Section II and IV referred to as *Worst Case*.

A different approach is proposed in [3] where the variance of a flow bandwidth requirements is proposed as a statistic to describe the application requirements. The authors further extend this method in [4] where they take into account the predicted fraction of packets delayed above a threshold. However, there is no proof that variance is a good descriptor for all traffic types.

In [5] a fuzzy-logic based controller is employed to predict the blocking probability of a particular flow. The authors claim that the varying nature of real time applications can be taken into consideration by a 'rule-based' controller. However, a validation of such controller against diverse types of traffic is not provided. Finally, in [6] an accurate admission control algorithm for video flows is proposed which takes into account both throughput and delay requirements. However, as we will see in Section II for the approach referred to as *Diophantine*, it can not be used in practice due to its computational load and therefore, an alternative is needed.

In this paper we design and evaluate an admission control algorithm, *E-Diophantine*, which aims at determining the feasibility of admitting a new flow into the system as accurately as possible while minimizing the computational load required for this operation. The rest of the paper is structured as follows. In Section II we propose a model for resource reservations in WiMAX networks and review potential solutions available in the literature. Our proposed E-Diophantine approach is described in Section III, providing its formal mathematical foundations and its accuracy and computational load compared to the alternative approaches presented in Section II. In Section IV we evaluate the different solutions considered using OPNET's WiMAX simulator in a realistic scenario. Finally, Section V summarizes our findings and concludes the paper.

II. ADMISSION CONTROL FOR WIMAX NETWORKS

WiMAX networks support QoS reservation of resources by allowing a new flow to apply for admittance in the system through a Dynamic Service Addition REQuest message (DSA-REQ). Such requests contain a QoS parameter set which includes different mandatory information depending on the service type. Table I summarizes the required QoS parameter set per data delivery service according to the IEEE 802.16 standard [1]. A similar set of parameters is required for scheduling services.

Based on these parameters, for each reservation i a minimum set of QoS requirements can be derived for a service as: given a starting time t_i, a certain amount of capacity B_i (bits) should be reserved periodically for transmitting flow's i data within a time interval T_i. Relevant examples of other wireless technologies which support reservation of resources in a similar way are 3G networks for cellular technologies and 802.11e HCCA for Wireless Local Area Networks.

Considering a new reservation i requesting acceptance in the system, an admission control algorithm has to evaluate whether

	UGS	ERT-VR	RT-VR	NRT-VR	BE
Min. Resv. Tr. Rate (MRTR)	x	x	x	x	
Max. Sust. Tr. Rate (MSTR)		x	x	x	x
SDU size	x				
Maximum Latency	x	x	x		
Tolerated Jitter	x	x			
Traffic Priority		x		x	x
Req./Trans. Policy	x	x	x	x	x

TABLE I

REQUIRED QOS PARAMETERS PER DATA DELIVERY SERVICE

there is enough capacity to admit the new reservation while still honoring the QoS of reservations already accepted. Such a resource resevation request can be modeled as a periodic discrete sequence of Kronecker deltas with amplitude B_i in the following way

$$B_i \cdot \delta_{t_i + n \cdot T_i}(t) = \begin{cases} B_i & \text{if } t = t_i + n \cdot T_i ; \ n \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Assuming a WiMAX system with a capacity available for data with QoS requirements C_{av} and N reservations already granted, a new reservation i can be accepted in the network if the following condition is met[1]

$$max(A(t)) \leq C_{av} \quad (2)$$

where $A(t)$ corresponds to the aggregation, as a function of time, of the reservations of the N flows already in the system plus the one requesting admittance.

A. Worst Case

In order to determine $max(A(t))$ different approaches can be considered. The easiest but more pessimistic approximation, hereinafter referred to as *Worst Case*, would be to assume that all admitted reservations need to be served simultaneously, i.e., without taking into account the time at which flows actually need to be served. The following equation corresponds to the *Worst Case* approximation of $A(t)$.

$$A_{worst\_case} = \sum_{i=1}^{N+1} B_i \quad (3)$$

Such an approach is similar to the one described in [2] and, as we will show in Section III-D and IV, it might result in a large portion of available capacity being underutilized.

B. Heuristic

An accurate solution for $max(A(t))$ can be obtained by computing all values of $A(t)$ within a T_{LCM} period. Note that since $A(t)$ is composed of *N+1* periodic reservations, its period T_{LCM} corresponds to the Least Common Multiple (LCM) of the periods of the reservations in the system plus the one under consideration. This approach will be referred in the rest of the paper as *Heuristic*. The following equation corresponds to the *Heuristic* computation of $A(t)$.

$$A_{heuristic}(t) = \sum_{i=1}^{N+1} B_i \cdot \delta_{t_i + n \cdot T_i}(t) \quad (4)$$

The *Heuristic* approach though has a dependence with the LCM of the reservations in the system which, depending on the granularity allowed, might increase exponentially with the number of reservations and thus, become too expensive in computational terms. Therefore, such a solution might not be feasible in practice unless a limitation in the granularity of periods is imposed, as it will be shown in Section III-D and IV.

C. Diophantine

In order to remove the LCM dependency with the *Heuristic* approach, another solution is considered based on *Diophantine*[2] theory which, in general, deals with indeterminate polynomial equations that allows variables to be integers only. In the rest of the paper this approach will be referred to as *Diophantine* and, as indicated in Section I, it has already been considered as a solution for admission control in WiMAX networks [6].

We define the *Diophantine* solution as follows. Considering a flow already accepted in the system described with the resource reservation $B_i \cdot \delta_{t_i + n_i \cdot T_i}(t)$ and a new flow requesting admittance characterized by $B_j \cdot \delta_{t_j + n_j \cdot T_j}(t)$, the maximum resource requirement, $B_i + B_j$, will occur for the set of n_i and n_j combinations which fulfill

$$\{t_i + n_i \cdot T_i = t_j + n_j \cdot T_j\} \quad (5)$$

$$\text{where } n_i \text{ and } n_j \ \in \mathbb{Z}$$

In order to find the set of solutions for n_i and n_j, hereinafter referred to as *set of intersections*, condition 5 can be expressed as a linear diophantine equation with two variables in the following way

$$\{n_i \cdot T_i - n_j \cdot T_j = t_j - t_i\} \quad (6)$$

Then, based on the linear diophantine equations theory, we know that there will be a set of integer solutions for n_i and n_j if

$$\frac{t_j - t_i}{d} \in \mathbb{Z} \quad (7)$$

where $d = gcd(T_i, T_j)$ and gcd stands for greatest commom divisor.

When the previous condition holds, the set of solutions corresponding to a specific pair of reservations can be found with the extended Euclidean algorithm which will find a and b such that

$$a \cdot T_i + b \cdot T_j = d \quad (8)$$

$$\text{where } a \text{ and } b \ \in \mathbb{Z}$$

By applying the *Diophantine* solution to all pairs of reservations in the system, as well as to their found solutions in a recursive manner, an exact solution for $A(t)$ can be found which is independent of the LCM length.

---

[1]Note that C_{av} does not necessarily have to correspond to the actual available capacity but could be a different value based on a specific operator policy.

[2]Diophantine equations are named after Diophantus of Alexandria, an Hellenistic mathematician of the 3rd century who studied such equations.

The *Diophantine* solution though requires to compute the gcd for all pairs of reservations in the system as well as the sets of intersections found. As a result, its computational complexity increases significantly as the number of reservations grow and, as in the case of the *Heuristic* solution, it might become unfeasible in practice.

III. E-DIOPHANTINE

Based on the performance issues identified for the *Worst Case*, *Heuristic* and *Diophantine* solutions, an enhancement of the *Diophantine* approach is proposed, hereinafter referred to as *E-Diophantine*. The objective is to achieve the same accuracy as the *Diophantine* solution when estimating the aggregated allocated maximum capacity but at a lower computational cost.

The *E-Diophantine* solution proposed consists in first, exactly as in the *Diophantine* case, finding the sets of intersections between all pair of reservations under consideration. After this step, instead of repeating the process in a recursive manner for all sets of intersections found, the results are structured in a matrix of intersections of reservations as the one shown in Table II for a 10 reservations example. Based on this matrix of intersections, the rest of the sets of intersections between the solutions found are derived based on the information obtained regarding the reservations involved in each intersection. In the following we provide the theorems and proofs that enable the designed *E-Diophantine* algorithm.

Reservations	1	2	3	4	5	6	7	8	9	10
1	1	0	1	1	1	1	1	1	0	1
2	0	1	1	1	1	1	1	1	0	1
3	1	1	1	1	0	0	0	1	1	0
4	1	1	1	1	1	1	1	1	1	1
5	1	1	0	1	1	0	1	1	1	0
6	1	1	0	1	0	1	0	1	1	0
7	1	1	0	1	1	0	1	1	1	0
8	1	1	1	1	1	1	1	1	1	1
9	0	0	1	1	1	1	1	1	1	1
10	1	1	0	1	0	0	0	1	1	1

TABLE II

EXAMPLE OF MATRIX OF INTERSECTIONS OF RESERVATIONS

A. Intersection of 2 Sets of Intersections of 2 Reservations

Theorem 1: For any pair of sets of intersections of 2 reservations found, they will intersect if both solutions have one reservation in common and the other two reservations intersect between each other

Proof: Consider that for reservations i and j a set of intersections exists defined as

$$\{t_{ij} + n_{ij} \cdot T_{ij}\} \tag{9}$$

$$\text{where } T_{ij} = lcm(T_i, T_j) \text{ and } n_{ij} \in \mathbb{Z}$$

such that the smallest n_i and $n_j \in \mathbb{Z}$ satisfy

$$t_i + n_i^{min} \cdot T_i = t_j + n_j^{min} \cdot T_j \equiv t_{ij} \tag{10}$$

Then, consider another set of intersections for reservations j and k defined as

$$\{t_{jk} + n_{jk} \cdot T_{jk}\} \tag{11}$$

A set of intersections between both intersections sets found will exist if a set of n_{ij} and $n_{jk} \in \mathbb{Z}$ such that

$$\{t_{ij} + n_{ij} \cdot T_{ij} = t_{jk} + n_{jk} \cdot T_{jk}\} \tag{12}$$

Considering that t_{ij} and t_{jk} can be expressed as $t_j + n_j^{min} \cdot T_j$ and $t_j + n_j'^{min} \cdot T_j$ respectively, we can express Eq. 12 as follows

$$\left\{ n_j^{min} + n_{ij} \frac{T_i}{gcd(T_i, T_j)} = n_j'^{min} + n_{jk} \frac{T_k}{gcd(T_j, T_k)} \right\} \tag{13}$$

Then, since

$$\left\{ n_j^{min} + n_{ij} \frac{T_i}{gcd(T_i, T_j)} \right\} \subseteq \{t_i + n_i \cdot T_i\} \tag{14}$$

and

$$\left\{ n_j'^{min} + n_{jk} \frac{T_k}{gcd(T_j, T_k)} \right\} \subseteq \{t_k + n_k \cdot T_k\} \tag{15}$$

we can affirm that a solution will exist for n_{ij} and $n_{jk} \in \mathbb{Z}$ such that the condition in Eq. 13 holds if reservations i and k intersect. The resulting set of intersections for reservations i, j and k would be then defined as

$$\{t_{ijk} + n_{ijk} \cdot T_{ijk}\} \tag{16}$$

$$\text{where } T_{ijk} = lcm(T_i, T_j, T_k) \text{ and } n_{ijk} \in \mathbb{Z}$$

■

B. Intersection of N+1 Sets of Intersections

Theorem 2: For any set of intersections of N sets of intersections found, it will intersect with another set of intersections if and only if all reservations involved in both sets of intersections intersect with each other

Proof: Assuming a set of intersections of N sets of intersections defined as[3]

$$\{t_{1-N} + n_{1-N} \cdot T_{1-N}\} \equiv I_N \tag{17}$$

For a set of intersections $t_{N+1} + n_{N+1} \cdot T_{N+1} \equiv I_{N+1}$ to intersect with I_N, a set of n_{1-N} and $n_{N+1} \in \mathbb{Z}$ should exist such that

$$\{t_{1-N} + n_{1-N} \cdot T_{1-N} = t_{N+1} + n_{N+1} \cdot T_{N+1}\} \tag{18}$$

Considering that

$$I_N = \{t_1 + n_1 \cdot T_1\} \cap ... \cap \{t_N + n_N \cdot T_N\} \tag{19}$$

Then, the set of intersections I_{N+1} will intersect with I_N if and only if

$$I_{N+1} \cap \{t_1 + n_1 \cdot T_1\} \cap ... \cap \{t_N + n_N \cdot T_N\} \notin \emptyset \tag{20}$$

■

[3]Note that the notation for a set of intersections has been simplified for readability reasons such that a set of intersections involving several reservations is referred with a single subindex instead of with the indexes of the reservations involved.

C. E-Diophantine Algorithm

Algorithm 1 details the steps followed by the *E-Diophantine* solution. The first part of the algorithm, which finds the first sets of intersections, is identical to the *Diophantine* algorithm. Once the first sets of intersections has been obtained, a matrix of intersections is computed. This operation corresponds to the function $compute\_matrix\_inters(.)$ in Algorithm 1. Table II provides an example of a matrix of intersections found for a set of 10 reservations. Such matrix of intersections can be built by traversing the sets of intersections obtained.

Based on the matrix of intersections, the *E-Diophantine* algorithm finds the rest of additional intersections by traversing the matrix of intersections and discarding the non-possible solutions applying Theorems 1 and 2. This operation corresponds to the function $compute\_inters\_inters(.)$ and yields the so-called $solutions\_tree$. After obtaining the tree of solutions, the maximum resource requirement when accepting the new reservation can be obtained by applying B to the solutions found.

D. Algorithms Performance Comparison

In order to evaluate the performance differences between the *Worst Case*, *Heuristic*, *Diophantine* and *E-Diophantine* approaches, we implemented these algorithms in $Matlab$ and performed the following experiment. We considered a system with 10 to 100 reservations where, for each one, t_i and T_i were randomly drawn from a uniform distribution as $granularity \cdot random(1, 100/granularity)$. Three different granularity values were evaluated: 1, 5 and 10. For illustration purposes B_i was taken as 1 for all reservations. Figure 1 summarizes the results of the experiment after running a minimum of 100 seeds for each value.

In Figure 1(a) the difference between the estimated maximum number of resources required by each of the approaches can be observed. Note that for the *Worst Case* performance the granularity is not considered since it has no time dependence. Taking the *Diophantine* and *E-Diophantine* results as reference since they represent the exact solution, as expected, the *Worst Case* solution is the one presenting the largest difference to the actual values; reaching differences above 300% in some cases. Such a large estimation deviation to the actual value would obviously result in available resources being underutilized and thus, in a lower revenue for a network operator. In the *Heuristic* case, the smaller the granularity the larger the difference to the actual value due to a limitation in the maximum T_{LCM} value that can be considered in a real implementation (10^7 in our system). Furthermore, the estimation is below the actual value and therefore, its usage for admission control purposes could compromise the QoS guarantees in a network. On the other hand, the *E-Diophantine* estimation is always equal to the *Diophantine* one, as expected based on Theorems 1 and 2.

In Figure 1(b) the corresponding differences in computational load are shown, computed as the percentage of reduction achieved with respect to the *Heuristic* approach which is taken here as reference due to its implementation simplicity.[4] As it can be observed, the *Diophantine* solution, although exact, ex-

---

Algorithm 1 E-Diophantine algorithm to find out the maximum resource requirement for a new reservation r_{N+1} with starting time t_{N+1}, period T_{N+1} and requirement B_{N+1} considering the set of N reservations already accepted in the system with their corresponding starting times $t = (t_1 ... t_N)$, periods $T = (T_1 ... T_N)$ and requirements $B = (B_1 ... B_N)$

```
1:    Call executed for each new reservation request
2:    for i = 1 to N + 1 do
3:        for j = i + 1 to N + 1 do
4:            if solution_exists(tᵢ, tⱼ, Tᵢ, Tⱼ) then
5:                intersections ← find_inters_dioph(tᵢ, tⱼ, Tᵢ, Tⱼ)
6:            end if
7:        end for
8:    end for
9:    m_inters = compute_matrix_inters(intersections)
10:   for i = 1 to N + 1 do
11:       solutions_tree ← compute_inters_inters(m_inters, i)
12:   end for
13:   if find_maximum(solutions_tree, B) ≤ C_av then
14:       return accept_request(r_{N+1})
15:   else
16:       return reject_request(r_{N+1})
17:   end if
```

ceeds by far the computational load of the alternative solutions considered and thus, it would not be feasible in practice [5]. For the largest granularity considered (*granularity 10*), the *Heuristic* approach clearly outperforms in computational time the *E-Diophantine* solution with no loss of accuracy. However, as the granularity considered decreases, the *E-Diophantine* computational time reduction with respect to the *Heuristic* one increases, being in all cases above 100% and reaching a maximum of about 2700%.

Based on these results, we can conclude that the *E-Diophantine* approach is, among the considered options, the most appropriate solution when no limitation in the period between resource reservations is desired or can be imposed. However, if a limitation is possible such that the LCM does not increase exponentially when increasing the number of reservations, the *Heuristic* approach can be exact when estimating and better in computational time.

IV. PERFORMANCE EVALUATION & DISCUSSION

In the previous section we have analyzed the performance of the proposed *E-Diophantine* solution as compared to its alternatives considering a generic scenario. In this section, we complete this evaluation by using OPNET's WiMAX simulator [7] to consider additional elements in the performance comparison that could have an impact in the maximum resource requirement estimation of the different approaches. Examples of these elements are: wireless physical channel, Transport layer, Network layer, MAC layer, control plane signaling, realistic applications, QoS schedulers, number of subscriber stations, etc.

In order to achieve this, a scenario is setup according to Table III and consisting of one Base Station (BS) and five Subscriber Stations (SS) where each station is configured to send and receive traffic from their corresponding pair in the wired domain of its type of application, i.e., one station sends and receives Voice traffic (without silence suppression), a second station sends and receives Voice traffic (with silence suppression), a third one receives a Video stream, a fourth one does an FTP download and the last one does Web browsing. Then, the num-

---

[4] The *Worst Case* is not considered since its computational load is obviously negligible but, as shown in Figure 1(a), its estimation accuracy is very poor.

[5] For instance, in the 30 reservations case with granularity 10, the computation time in a 2 · Quad Core simulation server took >1000 seconds.
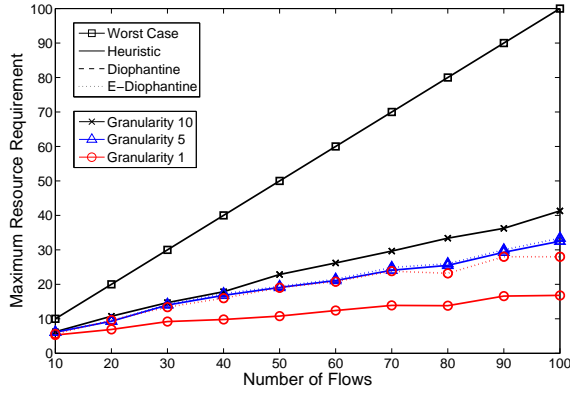
(a) Expected Maximum Resources Requirement
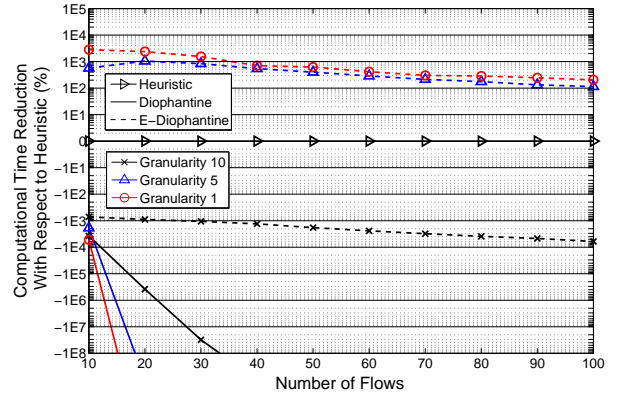
(b) Computational Time Reduction with Respect to Heuristic

Fig. 1. Comparison of Worst Case, Heuristic, Diophantine and E-Diophantine Approaches

ber of stations is increased in multiples of five stations up to 125 in total, always keeping the relation of $1/5$ of stations of each application type. The QoS scheduling policy chosen is Strict Priority applied first to fulfill the Minimum Reserved Traffic Rates (MRTRs) and then, the Maximum Sustained Traffic Rates (MSTRs). The length of the simulations performed is 120 seconds with a warm-up phase of 10 seconds. The number of seeds used to obtain average throughput values has been increased until their 95% confidence intervals did not overlap. In the case of the delay performance metric, the values represent the 95% percentile of the delay (CDF95) considering all simulation runs.

The configuration used for the different applications is detailed below:

• *Voice* G.711 Voice codec. Data rate: 64kb/s. Frame length: 20ms. Mapped to the *UGS* service in the downlink (BS → SS) and uplink direction.

• *Voice (silence suppression)* G.711 Voice codec. Data rate: 64kb/s. Frame length: 20ms. Talk spurt exponential with mean 0.35 seconds. Silence spurt exponential with mean 0.65 seconds. Mapped to *ERT-VR* in the downlink and to *ertPS* in the uplink.

• *Video* MPEG-4 real traces [8].Target rate: 450 kb/s. Peak: 4.6 Mb/s. Frame generation interval: 33ms. Mapped to *RT-VR* in the downlink and to *rtPS* in the uplink.

• *FTP* Download of a 20MB file. Mapped to *NRT-VR* in the downlink and to *nrtPS* in the uplink.

• *Web Browsing* Page interarrival time exponentially distributed with mean 60s. Page size 10KB plus 20 to 80 objects of a size uniformly distributed between 5KB and 10KB [9]. Mapped to the BE service both in the downlink and uplink direction.

Performance Results

In Figure 2(a) we show the peak and average throughput experienced in the downlink by the different application types as compared to the peak capacity estimations of the different approaches described in Sections II and III. The throughput of the differents applications is aggregated according to whether it is considered for admission control (*Premium* traffic: UGS+ERT-VR+RT-VR), or not (*Regular* traffic: NRT-VR+BE). Additionally, the average throughput of each single data delivery service belonging to the *Premium* group is provided as a reference.

WiMAX PHY Layer Config.	
Base Freq. (GHz)	2.5
Bandwidth (MHz)	10
Frame Duration (ms)	5
Symbol Duration (μs)	102.86
Number of Subcarriers	1024
DL Subfr. # Symbols	35
UL Subfr. # Symbols	12
DL Subfr. # Subch.	30
UL Subfr. # Subch.	35
# Data Subc./Subch	24
# SSs 64 QAM (3/4)	60%
# SSs 16 QAM (3/4)	30%
# SSs QPSK (1/2)	10%

Data Delivery Services		
UGS	MRTR: 80 Kb/s	
	Max. Lat: 20 ms	
ERT-VR	MRTR: 80 Kb/s	
	Max. Lat: 20 ms	
RT-VR	MSTR: 2 Mb/s	
	MRTR: 500 Kb/s	
	Max. Lat: 33 ms	

| E-Diophantine		
UGS	B_{UGS}: 1600 bits	
	T_{UGS}: 20 ms	
ERT-VR	B_{ERT}: 1600 bits	
	T_{ERT}: 20 ms	
RT-VR	B_{RT}: 16500 bits	
	T_{RT}: 33 ms	

TABLE III

PERFORMANCE EVALUATION PARAMETERS

From the performance results in Fig.2(a) the first remarkable result is that the peak of *Premium* traffic is in some cases above the peak estimated with the different admission control algorithms considered but the *Worst Case* one. The reason for this result is the 2Mb/s MSTR configured for RT-VR which allows video applications to get more than its 500 Kb/s MRTR if there is leftover capacity after serving all MRTRs. Note that the *Worst Case* estimation is too conservative and therefore, it will not be considered in the reminder of this section.

As the number of stations increases, the difference between the admission control estimations and the throughput peak of *Premium* traffic decreases. Note that the larger the amount of *Premium* traffic in the network, the lower the opportunities to go above the MRTR value. Eventually a point is reached where even the MRTR guarantees can not be satisfied, see crossing point between 20 and 25 stations per data delivery service. Moreover, as the number of flows in the system increases, the signaling overhead required for the DL-MAP increases as well, resulting in a lower *Premium* average throughput. For illustration purposes, an additional *E-diophantine* case has been added, *E-Dioph (1Mbps)*, where the MRTR for RT-VR has been configured to 1 Mb/s instead of 500 Kb/s. This case provides an example of how the admission control estimation would vary by allowing bursty traffic to transmit significantly above their average.
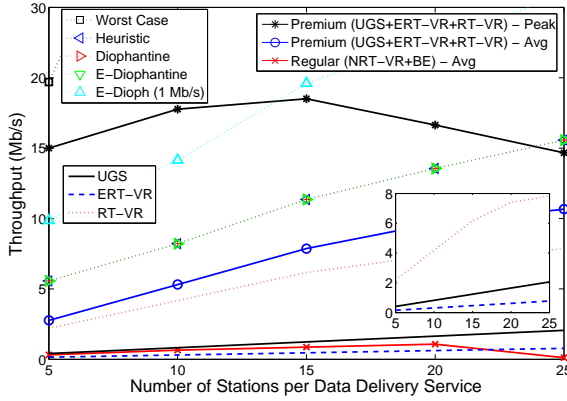
(a) Downlink Throughput
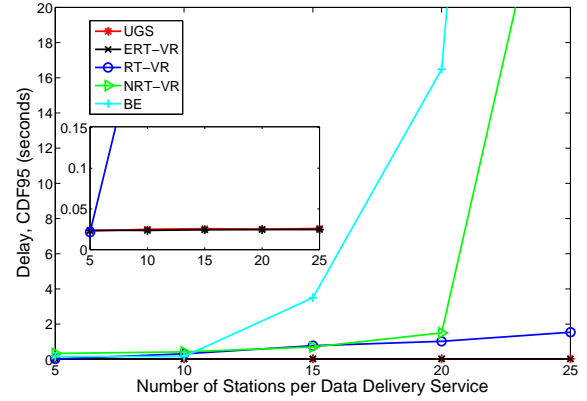
(b) Downlink Delay

Fig. 2. Network Performance for Data Delivery Services

With respect to the delay performance, the results are shown in Fig. 2(b). As expected, when the wireless resources become scarce, the delay experience degrades according to the traffic priority. In the case of RT-VR traffic, in contrast to UGS and ERT-VR, the delay experienced increases constantly. This is due to the performance metric chosen, 95% percentile of the delay (CDF95), which yields a close to worst case delay for each application traffic and thus, as the number of flows grows, it increasingly represents the Video peaks that can not be absorbed because there is not enough remaining capacity after serving all MRTRs. The delay performance of BE, which increases very rapidly, is due to the simple QoS scheduling policy used, Strict Priority, resulting in BE traffic being served only if the rest of the available traffic has already been served. Finally, both the NRT-VR and BE delay performance experience an extreme degradation after the 20 stations per data delivery service point. Note that this is where the estimation of the different admission control algorithms but the *Worst Case* crosses the *Premium* peak throughput and therefore, the probability for NRT-VR and BE traffic to be served decreases significantly.

The performance results corresponding to the uplink direction have been omitted due to space restrictions since, for the considered scenario, there is always enough capacity to satisfy the needs of all application flows and thus, no performance degradation is observed.

Based on these results we can conclude that the different solutions described in Sections II and III but the *Worst Case* could be effectively used to predict whether a new reservation should be admitted in the system. The selection of which algorithm would be more appropriate for a specific case should be taken considering the results presented in Section III-D.

V. SUMMARY & CONCLUSIONS

Networks with QoS guarantees require an admission control algorithm able to estimate the increase in allocated capacity needed if a new resource reservation is admitted. In this paper we have proposed the *E-Diophantine* solution, along with its mathematical foundations, and evaluated its benefits as compared to three alternative approaches, namely: *Worst Case*, *Heuristic* and *Diophantine*. The performance comparison comprised both accuracy and computational load analysis in a generic scenario as well as an evaluation using OPNET's WiMAX simulator in a realistic scenario.

The main conclusions that can be drawn from our results are: i) the *E-Diophantine* algorithm can be successfully used to predict the maximum allocated capacity demand of admitted QoS reservations in realistic scenarios; ii) the simpler *Heuristic* approach can outperform the *E-Diophantine* one in computational terms if limitations in the period between resource allocations can be imposed; iii) the larger the degree of flexibility allowed for defining the resource reservation periods, the larger the benefit of the *E-Diophantine* solution both in accuracy and computational load terms.

As future work we plan to extend the *E-Diophantine* algorithm to support WiMAX's multi-hop technology.

VI. ACKNOWLEDGMENTS

REFERENCES

[1] IEEE 802.16 Working Group, "IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Broadband Wireless Access Systems," IEEE Standard 802.16-2009, May 2009.

[2] H. Wang, W. Li, and D.P. Agrawal, "Dynamic Admission Control and QoS for 802.16 Wireless MAN," in *Proceeding of Wireless Telecommunications Symposium (WTS)*, Pomona, USA, April 2005.

[3] A. Teh and P. Pudney, "Efficient Admission Control Based on Predicted Traffic Characteristics," in *Proceeding of Personal Indoor Mobile Radio Communications (PIMRC)*, Athens, Greece, September 2007.

[4] A. Teh, A. Jayasuriya, and P. Pudney, "Admission Control in Wireless Infrastructure Networks Based on the Predicted Percentage of Delayed Packets," in *Proceeding of Asia-Pacific Conference on Communications (APCC)*, Tokyo, Japan, October 2008.

[5] S. Ghazal, Y.H. Aoul, J.B. Othman, and F. Nait-Abdessalam, "Applying a Self-Configuring Admission Control Algorithm in a New QoS Architecture for IEEE 802.16 Networks," in *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, Marrakech, Morocco, July 2008.

[6] O.Yang and J.Lu, "Call Admission Control and Scheduling Schemes with QoS support for Real-time Video Applications in IEEE 802.16 Networks," In Journal of Multimedia, May 2006.

[7] "OPNET Simulator," http://www.opnet.com.

[8] F.H.P. Fitzek and M.Reisslein, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation," IEEE Network, Vol. 15, No. 6, pages 40-54, November/December 2001.

[9] "Website Optimization," http://www.websiteoptimization.com/speed/tweak/.

[10] "EU FP7 Project Carrier Grade Mesh Networks (CARMEN), Nº 214994," http://www.ict-carmen.eu.