A comparative study of mixed traffic scenarios for different scheduling algorithms in WiMAX

Milad Alizadeh , Rudzidatul Akmam Dziyauddin* , Dritan Kaleshi and Angela Doufexi
Centre of Communication Research
Department of Electronics and Electrical Engineering
University of Bristol, UK
Email: {Milad.Alizadeh.10, Rudzi.Dziyauddin, Dritan.Kaleshi, A.Doufexi}@bristol.ac.uk

***Abstract*—WiMAX promises an advanced framework to support Quality-of-Service (QoS) requirements of different types of applications and scheduling is a key part in its QoS provisioning. The scheduling algorithms used in this paper are based on our proposed Greedy-Latency scheduler, a modified form of Greedy algorithm which can guarantee delay requirements of real-time applications while optimising the system throughput. Our study of TCP performance in WiMAX shows that unlike UDP traffic, there are fluctuations in TCP throughput even for low traffic loads. It is seen that employing Automatic Repeat reQuest (ARQ) and setting the right TCP window size are crucial for a stable optimal TCP performance. WiMAX QoS mechanism can successfully maintain the inter-class priority between TCP traffic in Best Effort (BE) class and UDP in higher priority Real-Time Polling Service (rtPS) class. For intra-class scenarios, it is observed that TCP flows in general need a protection mechanism as the UDP traffic tend to seize the channel. The proposed Greedy-Scheduler can provide better intra-class protection for TCP flows due to its packet dropping policy.**

***Index Terms*—IEEE 802.16, WiMAX, QoS, Scheduling, TCP, Greedy, Packet Dropping**

I. Introduction

Broadband wireless networks have evolved in recent years to offer a combination of voice and data services. The emergence of these multimedia services imposes a challenging QoS constraint on next generation of wireless networks such as WiMAX. WiMAX promises not only greater coverage and higher data rates, but also an advanced framework to support QoS requirements of different types of applications. The 802.16 standard currently defines five QoS classes which in order of priority are: Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Non-Real-Time Polling Service (nrtPS), Extended Real-Time Polling Service (ertPS) and Best Effort (BE).

Scheduling is a key part in QoS provisioning over a WiMAX network. The flexibility of WiMAX MAC layer allows separate scheduling schemes to be employed in each QoS service class. However, the 802.16 standard does not specify particular scheduling algorithms and leaves it to researchers and equipment makers to come up with their own implementations. Many scheduling algorithms have been proposed in the literature for WiMAX [1]–[3] and their performance have been well-studied for individual traffic types and QoS classes. It has been observed that Greedy scheduler can yield low average delays which implies it can potentially be used to serve delay sensitive applications [4]. However, the Greedy scheduler cannot guarantee any specific delay requirements and can lead to unfair resource allocations.

Our proposed scheduler is a channel-aware and latency-aware scheduling algorithm combined with a packet dropping policy which improves the fairness of the Greedy by taking into account the packet latency. The proposed scheduler not only serves users with good channel conditions but also prioritises packets in terms of latency requirements. Our study shows that some performance metrics of UDP traffic are improved by employing such scheduling algorithm. Our aim in this paper is to investigate how channel-aware and latency-aware scheduling and pre-emptive packet dropping impact WiMAX performance in mixed-traffic scenarios, with emphasis on TCP traffic. TCP has been traditionally designed for wired networks where it is assumed the underlying physical channel is very reliable. In wireless environments however, the channel can be very lossy and the effect of TCP in these environments can lead to unnecessary reduction in transmission rate and significant performance degradation. In this paper we study the performance of TCP protocol over a WiMAX network and identify the reciprocal effects of transport layer protocols and different scheduling schemes.

The rest of this paper is organised as follows. In Section II we introduce our proposed scheduling algorithm and discuss the packet dropping policy which is combined with other present downlink schedulers. Section III provides simulation model and parameters used in this paper. We present our simulation results and analysis in Section IV and summarise this paper in Section V.

II. Channel-Aware and Latency-aware Scheduling

A scheduler can use various parameters such as channel condition, packet latency or a packet dropping policy for making an optimal scheduling decision. Two types of attributes, namely, packet latency and packet deadline are often taken into consideration in schedulers designed to provide packet latency guarantees. The packet latency, often referred to as

* Rudzidatul Akmam is also with Universiti Teknologi Malaysia International Campus Kuala Lumpur (UTMKL), Jalan Semarak, 54100 Kuala Lumpur, Malaysia. (e-mail: rudzi@ic.utm.my).

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

packet delay, is the difference between the time a packet is sent from the source and the time it is arrived at the destination. A previous study also exploits the queuing time of a packet or the EDF (Earliest Deadline First) approach to compute the packet latency [5]. On the other hand, the packet deadline is often based on the admissible maximum latency, the sum of the admissible maximum latencies or other possible latency parameters such as packet arrival time at the BS [5]–[7]. When packets approach the maximum latency value, a dropping policy can have significant impact on the system performance.

Another challenge for serving users in an wireless network is the time-varying nature of the channel which means users are more likely to experience extreme and rapid signal variations during the time of transmission compared to wired networks. In many cases, due to uncorrelation between users' channels, a scheduler can exploit the diversity in channel quality to maximise the system throughput by serving the users with good channel conditions at every scheduling time.

In real communication systems, packets arrive at the BS with a certain probability distribution which leads to a variable queue length. It is therefore very important for the scheduling algorithm to consider the queues status as well as the channel conditions for an an optimal performance [8].

Very few studies [5], [9] consider all these three aspects (channel-aware scheduling, packet latency, and the dropping policy) together in the WiMAX system. Our Greedy-based scheduler proposed in the next section takes all these three aspects into account for an optimal channel-aware and delay-aware scheduling decision. From here on we refer to this combined scheduler as *Greedy-Latency*.

A. Greedy-Latency Scheduler

The utility function used by the Greedy-Latency guarantees the maximum admissible latency (T_k) while optimising the network throughput. Assuming that there are N active users with packets awaiting in queues at the BS, the utility function for the k-th user is given by:

$$U(d,t,\gamma) = \begin{cases} \arg\max_k (\frac{d_k}{T_k}\bar{\gamma_k}) & \text{for } \frac{d_k}{T_k} < 1 \\ \\ \text{Packet drop} & \text{for } \frac{d_k}{T_k} \geq 1 \end{cases} \quad (1)$$

$$T_k > 0 \ , \ d_k > 0 \ , \ \forall k \in \{1,2,3,\ldots,N\}$$

where $\bar{\gamma_k}$ and d_k denote user's average SNR and the Head-of-Line (HOL) packet latency respectively. The latter is computed as:

$$d_k = \text{Packet Scheduled Time - Packet Arrival Time in Queue} \quad (2)$$

The parameter d_k measures the delay the HOL packet waiting in the queue at the BS experiences until it is served. A more accurate expression can be defined using Equation 3 where the delay is measured from the time the HOL packet is sent from the Subscriber Station (SS) until it is scheduled at the BS:

$$d_k = \text{Packet Scheduled Time - Packet Sent Time} \quad (3)$$

The second approach is more accurate because it includes the propagation time from the SS to the BS in addition to the queuing time in the BS. However, this solution can be difficult to implement in practice as it requires accurate synchronisation between the base station and the subscriber station. In a single cell environment, which is the scope of this paper, the former approach is more realistic in measuring the HOL packet latency and is easier to implement. Therefore, in this study, the measurement of d_k follows Equation 2.

In real-time video streaming applications, the packets that exceed the latency requirements of the QoS are not useful when they arrive at their destinations. These packets are likely to be discarded by the receiver [10] and transmitting them is a waste of resources. Therefore, in Greedy-Latency, the HOL packet is considered for scheduling only if it is not expired, which is determined from the ratio of the HOL packet latency (d_k) to the allowed maximum latency (T_k). This value represents the *packet latency ratio* and is denoted by α. If α is less than 1, it is used in combination with user's average SNR to modify the normal scheduling decision of the Greedy algorithm; which is only based on SNR values. In the case that α is greater than 1, the packet is considered expired and must be dropped from the corresponding queue. The key benefit of this packet dropping policy is a more efficient usage of bandwidth.

The decision-making process of the Greedy-Latency algorithm can be summarised as follows: Firstly, the parameters d_k is computed using Equation 2 and α is calculated subsequently. If the computed α is less than 1, the utility function is formed using Equation 1, otherwise, the packet is considered expired and is dropped from the queue. Next, the calculated utility values for all non-dropped HOL packets are sorted in a descending order. Finally, the packet that has the maximum utility value is then served by the scheduler. This process is repeated for the next HOL packets. Note that FIFO queues are assumed in this study. The pseudocode of the Greedy-Latency algorithm is summarised in Table I.

TABLE I: Pseudocode of Greedy-Latency Scheduler

```
for every head-of-line packet (or active user) do
    Compute d_k from Equation 2;
    Compute α = d_k/T_k;
    if α < 1 then
        compute the utility function using Equation 1;
    else
        drop the packet from the queue;
    end if
end for
Sort the computed utility values in descending order;
Serve the HOL packet with maximum utility value;
```

Fig. 1 shows the utility function values of the Greedy-Latency algorithm for a range of different packet latency ratios and average SNRs, which in this paper are calculated on the

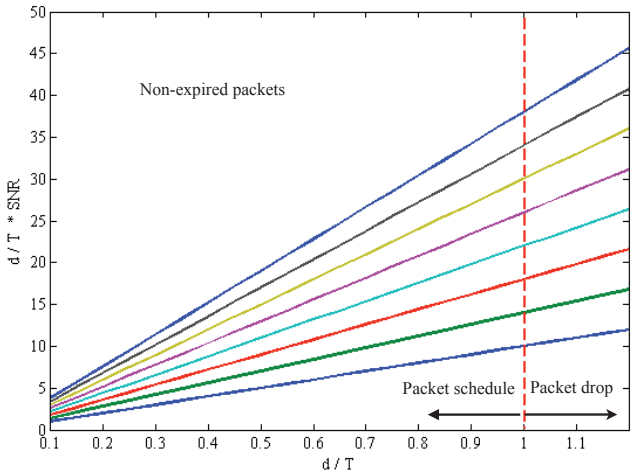

Fig. 1: Greedy-Latency scheduler Utility Function Range

moving average of a 3s window size. As an example, if there are two users in a cell and the first user has an average SNR of 30 dB with the packet latency ratio of 0.6 whilst the second user has an average SNR of 26 dB with the packet latency ratio of 0.9, the second user demonstrates a higher utility function value according to Fig. 1 and thus it is served first. The packet from the second user is closer to expiring compared to the first user which gives it higher priority despite having lower average SNR value. If both users have the same packet latency ratio, the first user who has the highest average SNR will be scheduled. This then leads to a Greedy scheduler behaviour. When both users have the same average SNRs, the Greedy-Latency prefers the user with the tighter packet latency; hence offers better service than the pure Greedy for such condition. In addition, with the introduction of packet latency ratio parameter, the proposed scheduler may aid cell-edge users in accessing the radio resources.

B. Combining Packet Dropping Policy with Other Schedulers

The same packet dropping (PD) policy introduced in the previous section can be applied to other schedulers to satisfy the required maximum latency and provide a fair comparison with the Greedy-Latency scheduler. In our study, we added the packet dropping to two leading channel-aware schedulers Proportional Fairness (represented by PF+PD) and Greedy (represented by Greedy+PD). Table II shows the pseudo code for combining the proposed packet dropping policy with other schedulers.

III. Simulation Model

The simulations in this paper have been performed using QualNet version 5.0. Table III summarises the system parameters used in this paper. All simulations are in single cell environment with either 10 or 25 users. Users are uniformly distributed and stationary. The link is only tested for uni-directional traffic on the downlink where all the resources allocations are done by the base station. Variable-Bit-Rate (VBR) packets with size of 300 bytes were chosen [9] for

TABLE II: Pseudocode for Combining Packet Dropping With Present Schedulers

```
for every head-of-line packet (or active user) do
    Compute d_k from Equation 2;
    Compute α = d_k / T_k;
    if α < 1 then
        compute the current scheduling algorithm;
    else
        drop the packet from the queue;
    end if
end for
```

UDP traffic to represent video streaming traffic in rtPS class. The TCP packet size in our simulations are 1500 bytes as suggested by Chrost *et al.* [11] to represent real-world greedy TCP sessions such as FTP. While all simulation times are 100 seconds, applications will not start transmitting until 15 seconds after the beginning of the simulation. The reason for choosing this setting is that subscriber stations take time to start operation and enter the network, therefore if applications start transmitting packets at 0 second, initial TCP packets will be garbled. This will trigger the TCP congestion avoidance mode. Delaying applications for 15 seconds will start TCP in slow start mode and quickly reach saturation.

TABLE III: Simulation Parameters for 10 and 25 Users

Configuration	Parameter	Value
PHY	Operating Frequency	2.5 GHz
	PHY Mode	OFDMA
	Duplexing Mode	TDD
	Channel Bandwidth	10 MHz
	FFT Size Data Subcarriers Sampling Factor Guard Interval	1024 720 8/7 1/8
	Propagation Model Propagation Limit Shadowing Model Shadowing Mean Fading	Two Ray Ground -110 dBm Log-Normal 8.9 Rayleigh
MAC	Frame Size	5 ms
	DL: UL Ratio	24:24
	Data Queue Size	500 KB
	Queue Algorithm	First-In-First-Out
	Link Adaptation and Fragmentation	Enabled
	ARQ	Disabled
Others	Transport and IP	UDP / IPv4
	Simulation Duration	100 seconds
Traffic Model	UDP Packet Size	300 Bytes
	TCP Packet Size	1500 Bytes
	UDP Traffic Type	VBR
	TCP Traffic Type	Super Application
Low UDP Traffic	Mean Packet Interval (25 Users)	5.22 ms
High UDP Traffic	Mean Packet Interval (25 Users)	3.64 ms
	Mean Packet Interval (10 Users)	2.20 ms

IV. Simulation Results and Analysis

A. Benchmarking TCP Performance

We begin our experiments by evaluating the performance of WiMAX for a TCP-only scenario in which all users have the same amount of traffic assigned to BE class. Before evaluating the performance of TCP, we first conduct a Load-Throughput simulation as suggested in [11] with UDP traffic. The result in Fig. 2 gives a basic insight into the impact of scheduling without considering TCP's elements such as congestion control mechanism. When the network is not heavily loaded, the use of different scheduling algorithms does not show any effect as there are sufficient resources in the network. As the load increases, schedulers comes into effect and different performances are observed for each scheduler. As expected, when the network is loaded closely to its capacity the Greedy scheduler gives the best channel utilisation followed by our proposed Greedy-Latency scheduler. For each scheduler the throughput reaches a maximum and then slightly decreases as the network enters saturation. Fig. 3 shows the results of the

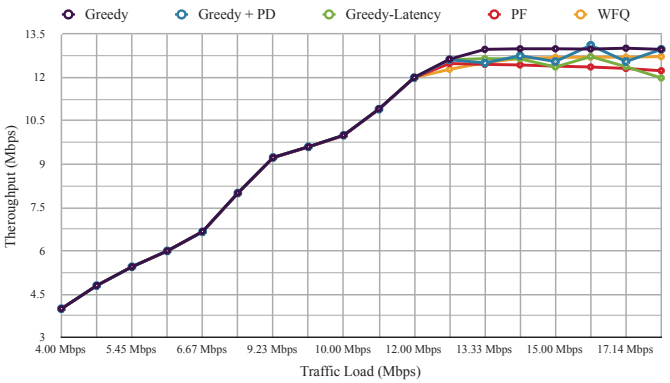

Fig. 2: Achievable throughput for UDP traffic in BE

same experiment for TCP traffic. This provides a benchmark in terms of maximum achievable throughput for the rest of the scenarios.

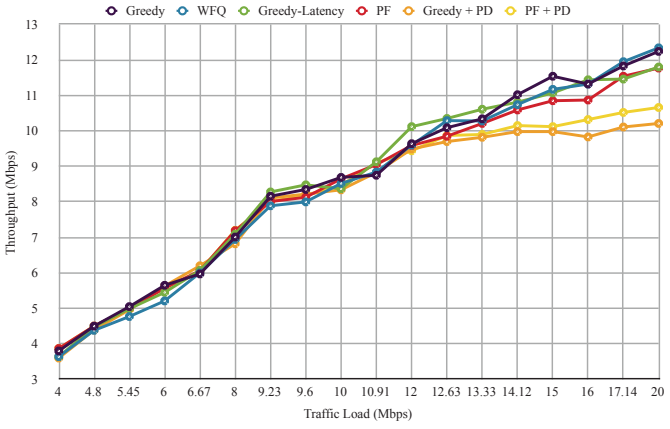

Fig. 3: Achievable throughput for TCP traffic in BE

1) Effect of ARQ: It can be seen that unlike UDP there are fluctuations in TCP throughput even for low traffic loads which differentiate schedulers from the beginning. This instability has been also observed in a commercial WiMAX test-bed [12] and even in WiMAX networks with high transmit powers [13] scenarios. It has been suggested that employing ARQ at the base station is critical for a stable performance of TCP throughput [14]. ARQ allows errors to be detected and packet retransmission to happen more efficiently, which results in a more reliable link seen by end-user applications. It has also been proposed that employing ARQ can compensate the differences in TCP variants [12].

2) TCP Variants: Numerous solutions have been proposed to optimise the performance of TCP over wireless links [15], [16] which have led to different wireless-friendly TCP variants. The TCP version used in this paper is Selective ACKnowledgment (TCP SACK) as suggested in [11], however, a selection of simulations scenarios were also repeated for other TCP versions (Lite, Reno, New Reno and Tahoe) to find out whether they make any meaningful difference. It was observed in our simulations, and also by others [12] in a commercial WiMAX test-bed, that all these TCP versions achieve more or less the same performance, though they achieve it in different ways.

3) Effect of TCP Window Size: It was observed in our simulations that setting the correct TCP windows sizes can significantly change the performance of TCP throughput. A small window size means the transmitter must stop sending packets and wait for the ACKs before being able to resume transmission. It has been suggested [13] that in order to have a stable throughput in WiMAX, the TCP window size should be between 85 KB and 256 KB. The value used in our simulations is 16 KB which is maximum allowed size in QualNet. Fig. 4 shows the throughput of WiMAX when the window size is changed from 512 bytes to 16 KB for WFQ scheduler. It can be seen that with a small window size the throughput is almost halved as the transmitter spends a long time waiting for AKCs feedback while a large window size utilises the channel near its theoretical capacity.

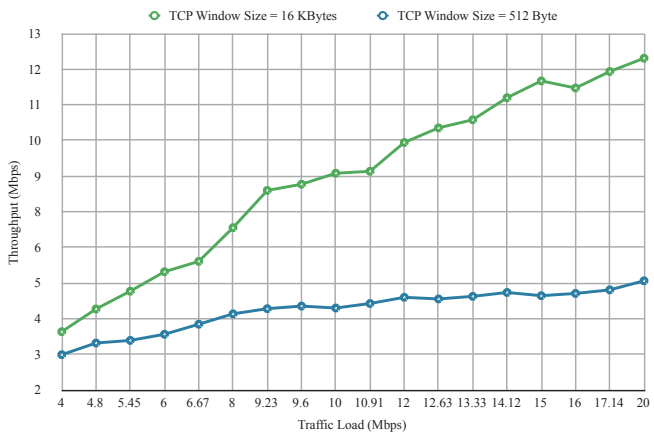

Fig. 4: Effects of TCP window size on throughput

B. Inter-class Scheduling for Mixed Traffic

In this section, we conduct a set of experiments in which all users have both UDP traffic in rtPS class, and TCP traffic in the BE. Since QualNet employs a strict priority scheduling between different classes, packets in BE are not served until all packets in the rtPS are scheduled. We therefore expect to see WiMAX protecting UDP sessions and allocating the remaining bandwidth to TCP flows. The throughput performance of

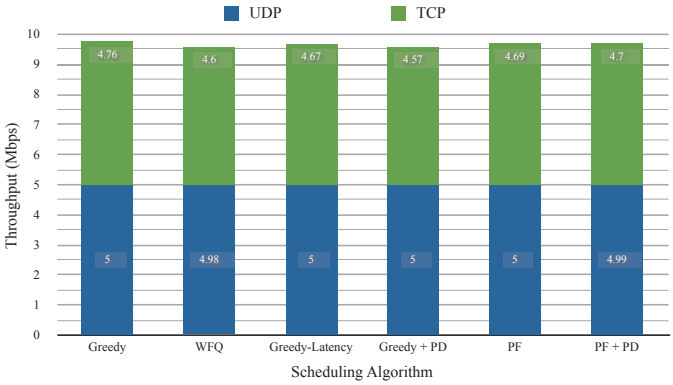

Fig. 5: System Throughput in Inter-class Mixed Traffic Aggregate Traffic Load 10 Mbps (5Mbps UDP, 5Mbps TCP)

different schedulers in Fig. 5 shows that the WiMAX QoS mechanism successfully maintains the priority between classes by allowing UDP flows to be scheduled first. The Greedy scheduler achieves the highest throughput among scheduling algorithms as expected followed by Greedy-Latency and WFQ. Fig. 6 shows that the QoS architecture successfully provides the delay requirements of the rtPS class (50 ms). For the TCP traffic, the Greedy scheduler produces the lowest average delay in all scenarios followed by Greedy-Latency and WFQ. The aggregate throughput in this experiment (10Mbps) is below the capacity of the network found in the previous section.

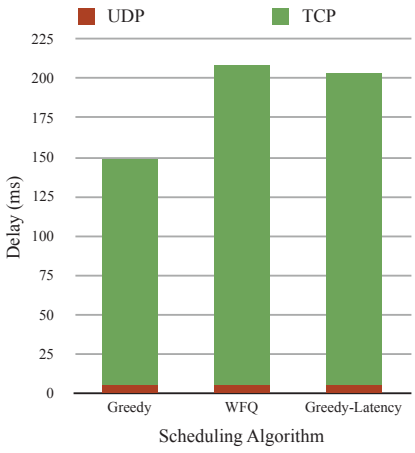

Fig. 6: Average Delay in Inter-class Mixed Traffic Aggregate Load 10 Mbps (5Mbps UDP, 5Mbps TCP)

To make sure the results reflect the impact of schedulers and are not influenced by not employing ARQ, both UDP and TCP traffic load were increased to a level that once UDP traffic is served by the scheduler, the remaining TCP traffic is well above the capacity of the network. Fig. 7 shows throughput performance of different schedulers for the inter-class scenario with 17Mbps aggregate load. As expected the Greedy sched-

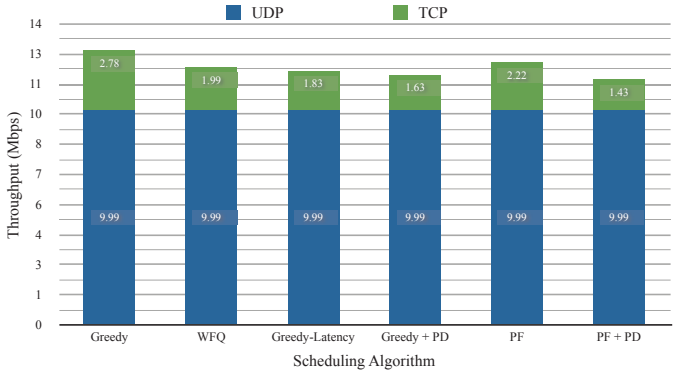

Fig. 7: System Throughput in Inter-Class Mixed Traffic Aggregate Load 17 Mbps (10Mbps UDP, 7Mbps TCP)

uler achieves the best performance, however, it is interesting to see that the channel-agnostic WFQ scheduler outperforms channel-aware schedulers such as Greedy-Latency and Pure Greedy with packet dropping policy. The reason lies in the congestion control mechanism in TCP. It has been suggested [14] that queue-aware schedulers such as WFQ or MLWDF can behave like Greedy schedulers when the network is congested, as is the case in our scenario. Giving priority to longer queues in a queue-aware scheduler encourages it to further increase its window size and potentially seize the channel [14] which can lead to an unfair resource allocation in congested networks.

C. Intra-class Scheduling for Mixed Traffic

While the majority of classic data services such as web browsing, email and file transfer are built upon TCP reliable connections, the emerging multimedia services are increasingly using UDP for transmission, which has no support for congestion control. One of the drawback of having more UDP traffic is unfairness for intra-class competing TCP flows [17], [18]. In this section we study the impact of different scheduling algorithms on mixed traffic scenarios in the rtPS class. Fig. 8 shows the system throughput for a heavily loaded system (10 Mbps UDP + 7 Mbps TCP). It can be seen that TCP flows in general need an intra-class protection mechanism as the UDP traffic tend to seize the channel. An intra-class hierarchical has been proposed to [18] to distinguish TCP and UDP within the same class, however, it can be seen that the proposed Greedy-Latency scheduler can provide a better protection for TCP flows due to its algorithm considering queue status as well as the user's channel conditions. When the traffic load of TCP is below the network capacity there is not a severe fairness issue as both traffic types get a fair share or network resources. This is shown in Fig. 9 for an aggregate load of 14 Mbps (7Mbps UDP, 7Mbps TCP)

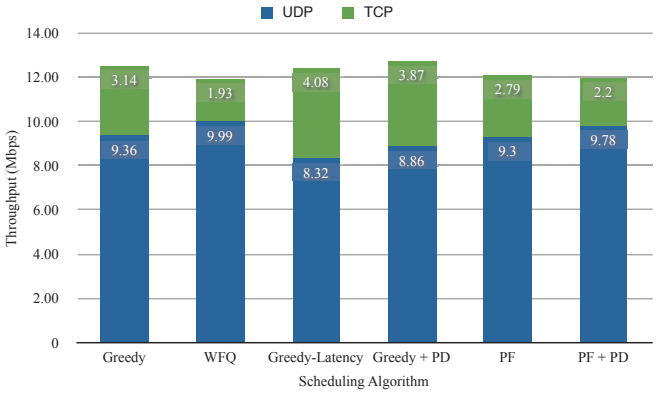

Fig. 8: System Throughput in Intra-Class Mixed Traffic Aggregate Load 17 Mbps (10Mbps UDP, 7Mbps TCP)

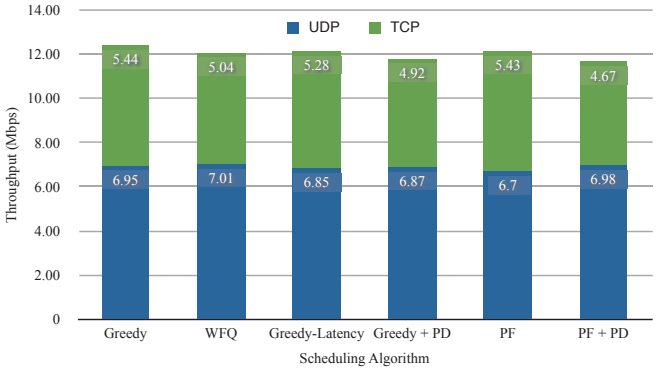

Fig. 9: System Throughput in Intra-Class Mixed Traffic Aggregate Load 14 Mbps (7Mbps UDP, 7Mbps TCP)

V. Conclusion

In this paper, we show that the flexibility of the WiMAX MAC layer in using class-specific schedulers is a key aspect for optimising the end-to-end performance of WiMAX. It is observed that the WiMAX QoS mechanism successfully maintains the priority between classes in mixed-traffic scenarios by scheduling UDP flows in rtPS class first. However, we argue that a specific TCP-friendly scheduler for the BE class can significantly improve the performance of TCP flows in the BE class.

It is observed that without employing ARQ, the throughput of TCP is not stable. It is also very important to ensure that the correct settings have been selected for the TCP window size in WiMAX. Significant degradation was observed in TCP throughput for insufficient window sizes.

Our results show that that flexibility of WiMAX MAC layer can be exploited to enhance the overall TCP performance in intra-class scenarios. This paper introduces the concept of a joint channel-aware and delay-aware scheduling and packet dropping to satisfy the admissible maximum latency whilst optimising the system throughput for WiMAX networks. It is seen that the Greedy-Latency scheduler can provide better protection for TCP flows against UDP traffic in intra-class scenarios.

References

[1] C. So-In, R. Jain, and A.-K. Tamimi, "Scheduling in ieee 802.16e mobile wimax networks: key issues and a survey," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 2, pp. 156 –171, february 2009.

[2] C. Valencia and T. Kunz, "Scheduling alternatives for mobile wimax end-to-end simulations and analysis," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 371–375.

[3] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking (Prentice Hall Communications Engineering and Emerging Technologies Series)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2007.

[4] M. Nicolaou, A. Doufexi, S. Armour, and Y. Sun, "Scheduling techniques for improving call capacity for voip traffic in mimo-ofdma networks," in *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, sept. 2009, pp. 1 –5.

[5] A. Lera, A. Molinaro, and S. Pizzi, "Channel-aware scheduling for qos and fairness provisioning in ieee 802.16/wimax broadband wireless access systems," *Network, IEEE*, vol. 21, no. 5, pp. 34 –41, sept.-oct. 2007.

[6] T. Ali-Yahiya, A.-L. Beylot, and G. Pujolle, "An adaptive cross-layer design for multiservice scheduling in ofdma based mobile wimax systems," *Computer Communications*, vol. 32, no. 3, pp. 531 – 539, 2009, adaptive Multicarrier Communications and Networks.

[7] H. Wang and L. Dittmann, "Downlink resource management for qos scheduling in ieee 802.16 wimax networks," *Comput. Commun.*, vol. 33, pp. 940–953, May 2010.

[8] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, no. 02, pp. 191–217, 2004.

[9] N. A. Ali, P. Dhrona, and H. Hassanein, "A performance study of uplink scheduling algorithms in point-to-multipoint wimax networks," *Comput. Commun.*, vol. 32, pp. 511–521, February 2009.

[10] P. Pahalawatta, R. Berry, T. Pappas, and A. Katsaggelos, "Content-aware resource allocation and packet scheduling for video transmission over wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 4, pp. 749 –759, may 2007.

[11] L. Chrost and A. Brachman, "Towards a common benchmark in wimax environment," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*, may 2009.

[12] E. Halepovic, Q. Wu, C. Williamson, and M. Ghaderi, "Tcp over wimax: A measurement study," in *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*, sept. 2008, pp. 1 –10.

[13] F. Zarrar Yousaf, K. Daniel, and W. Wietfeld, in *Analyzing the Throughput and QoS Performance of WiMAX Link in an Urban Environment, WIMAX New Developments, Upena D Dalal and Y P Kosta*.

[14] X. Yang, M. Venkatachalam, and S. Mohanty, "Exploiting the mac layer flexibility of wimax to systematically enhance tcp performance," in *Mobile WiMAX Symposium, 2007. IEEE*, march 2007, pp. 60 –65.

[15] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving tcp performance over wireless links," *Networking, IEEE/ACM Transactions on*, vol. 5, no. 6, pp. 756 –769, dec 1997.

[16] B. Bakshi, P. Krishna, N. Vaidya, and D. Pradhan, "Improving performance of tcp over wireless networks," in *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*, may 1997, pp. 365 –373.

[17] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *Networking, IEEE/ACM Transactions on*, vol. 7, no. 4, pp. 458 –472, aug 1999.

[18] K. Balakrishnan, "Video streaming optimization and tcp throughput protection over last mile broadband wireless access networks," Master's thesis, Banglore International Institute of Information Technology,.