Multimedia Traffic Scheduling in DVB-S2 Networks with Mobile Users

Gorry Fairhurst*, Giovanni Giambene†, Samuele Giannetti†, Cristina Párraga‡, Aduwati Sali§

*University of Aberdeen, Department of Engineering, Fraser Noble Building, Aberdeen AB24 3UE, UK
†University of Siena, Via Roma, 56, I-53100 Siena, Italy
‡German Aerospace Center (DLR), Oberpfaffenhofen, P.O. Box 11 16, D-82230, Wessling, Germany
§Centre for Communication Systems Research, University of Surrey, Guildford, Surrey GU2 7XH, UK

***Abstract*—DVB-S2 is a second-generation standard for satellite transmissions, developed to improve the performance of DVB-S. Recently, research efforts have been devoted to extend this standard for mobile usage. In this paper we consider a scenario composed of a Ku-band GEO bent-pipe satellite, a gateway and some mobile users. The interest is here in investigating the performance of a scheduling scheme to manage DVB-S2 resources on the forward channel for the delivery of video CBR and FTP traffic flows to mobile users. A novel scheduler scheme, called UBMT, has been proposed that aims to manage CBR packet transmission deadlines while maximizing the aggregated goodput. Simulation runs have permitted to validate the optimal performance achieved by UBMT in comparison with other schemes. The impact of different TCP versions has been investigated as well showing the good results obtained with the SACK option.**

I. Introduction

IP-based broadband satellite communications represent an important part of the global telecommunications market. The current research interest is on the provision of differentiated *Quality of Service* (QoS) levels and the efficient utilization of radio resources. Towards this end, great importance is assumed by the *Digital Video Broadcasting - Satellite version 2* (DVB-S2) standard [1] that supports *Adaptive Coding and Modulation* (ACM) for interactive services [2]; ACM permits to change dynamically *Modulation and Coding* (ModCod) on the basis of channel conditions. The sender site dynamically acquires information on the receiver channel conditions by means of a return link. Four modulations can be used for the forward link [3]: QPSK, 8PSK, 16APSK, and 32APSK. Moreover, DVB-S2 exploits *Forward Error Correction* (FEC) encoding that is obtained by the concatenation of *Bose-Chaudhuri-Hocquenghem* (BCH) outer code and *Low Density Parity Check* (LDPC) inner codes. The selected LDPC codes operate with rates of 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9 and 9/10, depending on the adopted modulation and the system requirements. Considering the above-mentioned four modulations and the eleven code rates, 28 ModCod combinations are considered in the standard.

The DVB-S2 standard was specifically designed for fixed terminals in Ku and Ka bands; in such an environment, the propagation issues to cope with are mainly the variable fading due to water absorption and scintillation. This work investigates a scenario (see Figure 1) with a *Geostationary Earth Orbit* (GEO) bent-pipe satellite, a network gateway & feeder and two classes of mobile users (real-time video users and non real-time FTP users). The gateway, that also includes the *Network Control Centre* (NCC), is used to access the Internet. The path from the gateway to the satellite and then to the users is based on DVB-S2, while the return path adopts the *DVB Return Channel via Satellite* (DVB-RCS) standard.

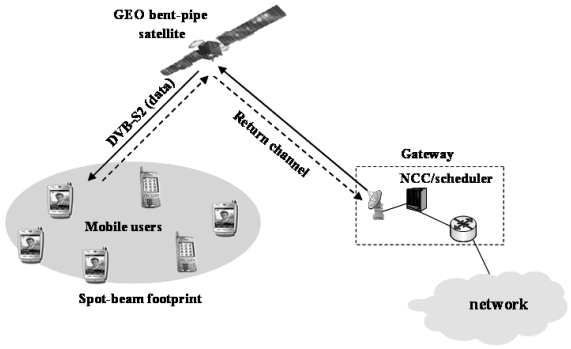

Fig. 1. System architecture and scenario for DVB-S2 with mobile users. Both FTP traffic and CBR video streaming (or video conference) downstream traffic flows are considered.

II. DVB-S2 PHY Modelling

We consider the ACM method of DVB-S2 to counteract the time-varying propagation conditions in Ku-band, especially critical for the mobile environment. In particular, each receiver monitors its own channel conditions and reports this information to the transmitter; the transmitter then selects an appropriate ModCod for each frame transmission. This capability to adapt the transmission mode to current channel conditions permits an efficient use of the satellite bandwidth and guarantees to control the occurrence of errors. The ModCod adaptation is implemented by applying the selected coding rate and modulation to the *Base Band Frame* (BBFrame). The simplified *Physical Layer frame* (PL frame) generation process is depicted in Figure 2. Note that only one ModCod can be used for the transmission of a BBFrame (and related conveyed IP packets). The BBFrame is encoded according to the selected ModCod to form a FECFrame of fixed length n_{FEC} (either normal, 64800 bits, or short, 16200 bits). This means that the maximum length of the BBFrame data field (K_{bch} in Figure 2) depends on the selected coding rate. The BBFrame must

978-1-4244-1948-7/08/$25.00

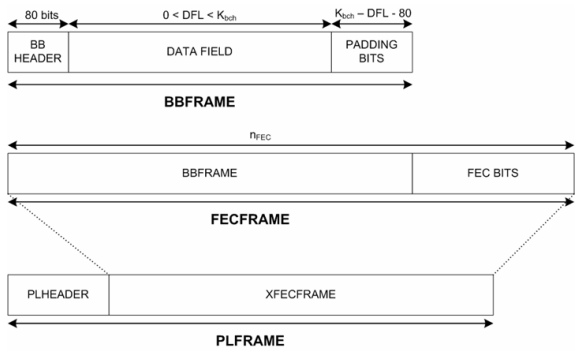

Fig. 2. DVB-S2 (simplified) physical frame generation process.

TABLE I
ADOPTED MODCODS FROM THE STANDARD (THE NUMBERS USED IN THIS WORK ARE DENOTED BY AN ASTERISK) AND RELATED CHARACTERISTICS.

ModCod	Capacity [Mbit/s]	E_b/N_0 QEF threshold	PL frame duration
1 → 1*	12.5	0.624 dB	1.3 ms
6 → 2*	33.5	1.765 dB	1.3 ms
11 → 3*	45	3.773 dB	1.3 ms
20 → 4*	80	5.903 dB	0.652 ms

be padded if the contained user data do not reach the *Data Field Length* (DFL); this occurs if the applied encapsulation method does not allow packet fragmentation. Later on, the FECFrame is modulated according to the selected ModCod to form an XFECFrame. Upon addition of the corresponding header (PLHEADER), the resulting PL frame is then transmitted in time division multiplexing way. The ModCod can dynamically change on a frame-by-frame basis, increasing or decreasing the effective throughput achieved. The ModCod selection given the channel quality estimations reported by the receivers is based on the following trade-off: maximizing the effective throughput without exceeding a *Frame Error Rate* (FER) of 10^{-7} (i.e., *Quasi Error-Free*, QEF, operation). To achieve this performance, the standard provides the minimum required (threshold value) *Energy per bit-to-Noise spectral density* (E_b/N_0) to guarantee QEF transmissions with that ModCod.

To simplify the implementation in our simulator, four non-overlapping transmission ModCods have been selected from those specified by DVB-S2, as shown in Table I. The DVB-S2 symbol rate is fixed and equal to 25 Msymbols/s and the FECFrame length is 64800 bits.

Simulations of a Ku-band channel model were conducted to characterize the temporal behavior of the channel. E_b/N_0 traces were generated and used to evaluate different fade mitigation techniques. We have used a channel trace of 4000 s (with E_b/N_0 ranging from 0.5 to 6.3 dB), where signals are transmitted by a horizontal-polarization GEO satellite antenna and received by a low gain flat antenna with 19 dBi gain. The propagation scenario considered is a suburban environment (land-mobile users) composed of obstacles such as family houses or villas with gardens [4]. The speed is set to be 30 km/h, but includes sudden changes in the motion direction. To simulate this environment, a two-state Markov model has been developed, which alternates between open (line-of-sight) and fading (shadowed and blocked) states. Different users employ the channel trace with different time offsets (i.e., the starting instants are uniformly distributed over the trace) so that users experience independent channel conditions. With a sufficiently number of terminals, we may cover all the portion of the channel trace with a 600 s simulation run.

The high satellite propagation delay associated with a GEO-based satellite system can generate a misalignment between the current E_b/N_0 value at the receiver and the ModCod used by the scheduler for the transmission. At least one *Round Trip Propagation Delay* (RTPD equal to 560 ms) passes from the measurement that determines the ModCod value to the instant when a PL frame is received by the terminal using the selected ModCod. A possible approach to reduce the misalignment effects is to adopt a channel estimation scheme that predicts future channel behavior. An idealized scheme is considered here for channel estimation: we assume that channel estimation at time t permits to determine correctly the future E_b/N_0 value [in dB] after RTPD, $E_b/N_0(t + RTPD)$, with probability p. This results in the following estimation scheme:

$$\frac{E_b}{N_0}(t)_{estimated} = (1-p)\frac{E_b}{N_0}(t) + p\frac{E_b}{N_0}(t + RTPD) \ . \quad (1)$$

Our Ku-band channel model (suburban scenario) has been used to estimate the p value of the channel estimation; we have obtained p = 0.824.

A protection margin h has been also adopted when selecting the ModCod, leading to the following expression [in dB]:

$$\left[\frac{E_b}{N_0}(t)\right]^* = \frac{E_b}{N_0}(t)_{estimated} - h \ . \quad (2)$$

The effect of introducing h is that it reduces the system capacity (conservative ModCod selection); therefore, the value of h must be carefully selected.

A preliminary study has been carried out for the selection of the h value considering only FTP traffic with TCP NewReno (impatient option) congestion control and a standard 64 kB window size. It resulted that h = 1 dB allows maximizing the single-user TCP goodput and we selected that value. The resulting ModCod probability distribution for h = 1 dB and p = 0.824 (suburban scenario) is shown in Figure 3, where we have considered that in case of outage (i.e., E_b/N_0 value below the threshold of ModCod = 1*, the most robust modulation and coding option) we select anyway ModCod = 1* for transmissions. We can thus derive the mean system capacity C_s by weighting the capacity values in Table I with the obtained probability distribution; the average capacity is C_s = 34.1 Mbit/s. Such capacity value does not account for the loss of efficiency due to the partial filling of BBFrames by IP packets. A frame loss event occurs if at the arrival instant t of the frame with a given ModCod j, $E_b/N_0(t)$ is below the threshold of ModCod j. Note that a frame may contain several IP packets

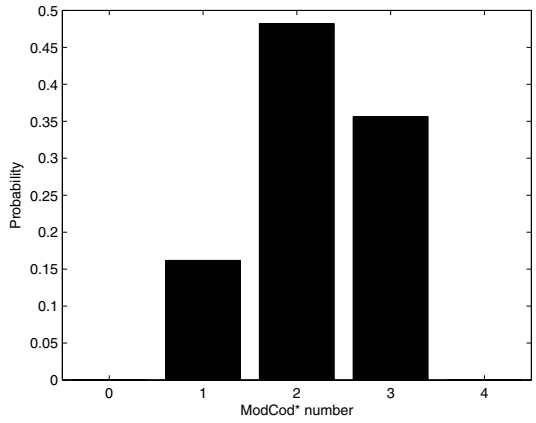

Fig. 3. Distribution for the use of the four ModCods with h = 1 dB and p = 0.824 for the suburban scenario (we have assumed that in case of outage, we select anyway ModCod = 1* for transmissions).

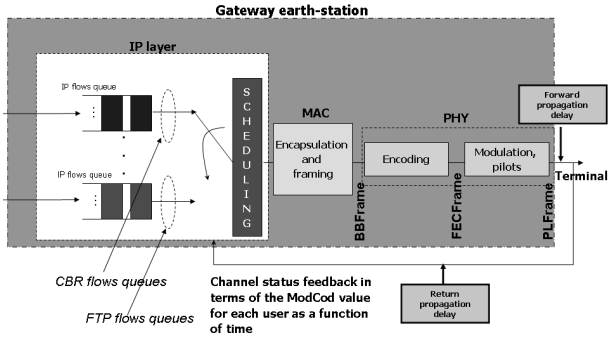

Fig. 4. Layer 3 multi-queue architecture, encapsulation and ModCod level.

destined to different users so that this loss check may give different outcomes on the same frame for distinct users, since they may experience at time t different channel conditions. Finally, the obtained mean IP packet loss rate for a single FTP flow case with h = 1 dB and p = 0.824 is about 0.5%.

III. Proposed Scheduling Strategy

The interest of this study is to schedule multimedia traffic flows (i.e., video CBR and FTP) on the DVB-S2 forward link. According to Figure 4, a per-flow queue architecture is envisaged at the IP level, where there is one queue at the gateway for each flow and flow-related QoS parameters are considered (e.g., packet deadlines for real-time flows). This queue architecture could be supported by using IntServ that permits a fine control of the QoS provided to each flow.

As for FTP traffic, elephant connections are used where each packet is formed of 1490 bytes. Each video stream is a CBR flow with IP packets of 502 bytes, representing video frames regularly generated every 13 ms (SIF format) with a corresponding bit-rate of 315 kbit/s. Each CBR packet is associated with a transmission deadline of 100 ms; if deadline expires, the packet is dropped from the transmission queue.

We consider that an IP packet is encapsulated using the *Generic Stream Encapsulation* (GSE) forming a MAC layer

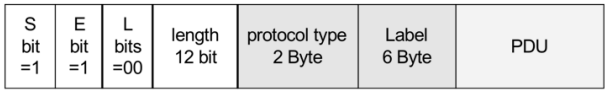

Fig. 5. GSE encapsulation format without IP fragmentation.

packet that is scheduled in a BBFrame. We do not consider here fragmentation at the IP or GSE levels (this aspect is left to a future study). Therefore, if the gateway cannot entirely fill in a BBFrame with IP packets, padding is employed for the remainder of the BBframe. In this case, GSE introduces a header of 10 bytes to each IP packet, with the header format shown in Figure 5. Hence, for each IP packet due to a video flow (or for each IP packet due to FTP), a MAC layer packet of 512 bytes (1500 bytes) has to be scheduled in a BBFrame.

Scheduling decisions are taken at the beginning of each new PL frame transmission, denoted in the following description as *scheduling instant*([1]). At each new scheduling instant, the scheduler determines how many packets can be transmitted for each traffic class using each ModCod and the related receiving terminal. Note that our method uses per-flow queuing, where there is one queue per traffic flow per user, so that at any one time, each queue corresponds to one ModCod, although several queues can be assigned to the same ModCod. If there are n IP packets that can be transmitted with ModCod j, they can be also transmitted with ModCod $j-1$, ModCod $j-2$, etc., but with a lower efficiency (i.e., a greater time is needed to transmit them than the minimum possible). Once a ModCod has been selected for the transmission of the next PL frame, there are many possible combinations of IP packets of the two types (i.e., 1500 byte long for FTP and 512 byte long for CBR) that can be allocated. These combinations also depend on both the current number of packets available for transmission in the different ModCods and the number of active CBR and FTP flows. Typically, there are many combinations we can choose from for ModCods at each new scheduling instant (*feasibility space* for allocations). Assuming an arbitrarily-high number of packets in both the FTP and CBR queues belonging to the four ModCods, we obtain the feasibility spaces for allocation for the 4 ModCods, as shown in Figure 6. Once a given ModCod $j*$ is selected with the related combination of n_{CBR} CBR packets and n_{FTP} FTP packets to be transmitted in the next BBFrame, there are different ways to choose the n_{CBR} CBR packets and n_{FTP} FTP packets respectively from those in CBR and FTP queues that can support ModCod $j*$ or higher ones. Packets from users supporting higher ModCods than $j*$ should be considered for transmission with ModCod $j*$ only when there are not enough packets belonging to ModCod $j*$ in the queues[2]. We consider selecting the n_{CBR} CBR packets and the n_{FTP} FTP packets from the related queues according to a differentiated forwarding scheme with prioritization given by

[1]From Table I, the PL frame duration varies depending on the ModCod.

[2]To design a work-conserving scheduler, achieving maximum resource utilization efficiency, we consider packets using ModCod values higher than $j*$ to avoid as much as possible leaving unused parts of the BBFrame.

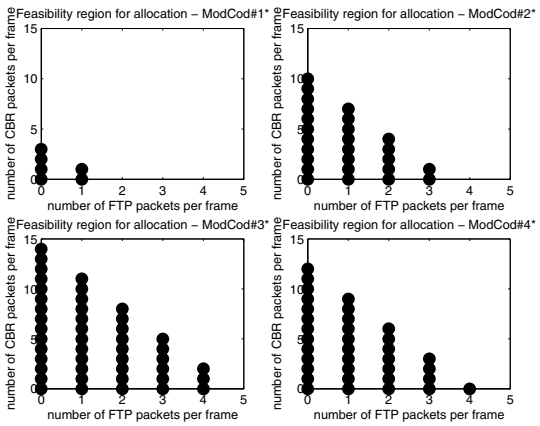

Fig. 6. Feasibility spaces for allocation for the 4 ModCods considered and having an arbitrarily-high number of packets in both FTP and CBR queues belonging to the four ModCods; bullets denote feasible allocations.

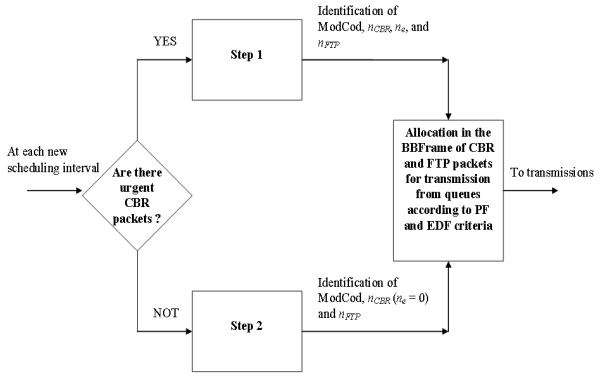

Fig. 7. Block diagram of the proposed UBMT scheduling algorithm.

the *Earliest Deadline First* (EDF) scheme for CBR packets and *Proportional Fairness* (PF) scheme for FTP packets (with sliding window equal to 1000 frames since TCP behavior is slow). These priority indexes are flow-based and applied at the level of IP packets. Note that depending on the length of IP packets combined to fill in a BBFrame with a given ModCod, the BBFrame can result only partly used (we do not consider here fragmentation) and this entails some inefficiency.

We propose a channel-aware IP packet scheduler that also considers the expedited delivery requirements for CBR packets. The rationale of our scheme is that we schedule CBR packets and FTP packets (IP packets going in one BBFrame) in order to minimize the CBR packet loss rate due to deadline expiration and, at the same time, in order to maximize the aggregated goodput with appropriate ModCod selection. Our proposed scheduler is called *Urgency-Based Maximum Throughput* (UBMT). This is an optimized scheme that is quite interesting since it can provide a good understanding of the best performance attainable for the two traffic classes. The idea is that the CBR packets that are subject to deadline are serviced exactly at their deadline times, except anticipating their service in case that there are unused resources in previous frames; we fill the gaps among CBR scheduled packets by means of FTP packets with the aim to maximize the resource utilization. At each new scheduling instant (transmission of next PL frame), we select the ModCod and the related packets to be served according to the two alternative steps below (see Figure 7).

- **Step 1**: *Selection of ModCod and number of CBR and FTP packets to be transmitted by the next frame in the presence of urgent CBR packets*: if at the scheduling instant (where we have to fill in the next BBFrame with IP packets) there is at least one CBR packet whose deadline is expiring (within the maximum next PL frame duration; see Table 1), we are forced to select ModCod $j*$ that permit to maximize the number of allocated CBR packets, n_e, with deadline expiring in this scheduling interval (i.e., to minimize the number of CBR packet losses due to deadline expiration). After selection of ModCod $j*$, we fill the remainder of the next BBFrame with other FTP or CBR packets seeking to maximize the throughput achieved with the BBFrame transmission, by considering packets with ModCod $j*$ or higher. At this point we have determined the ModCod $j*$ selected for the transmission of the next frame, the number of allocated CBR packets, n_{CBR}, and the number of allocated FTP packets, n_{FTP}. We now determine exactly which packets have to be serviced from the FTP and CBR traffic queues. Once the n_e expiring CBR packets are allocated, we service the CBR queues according to the ModCod and EDF order to select the remaining $n_{CBR}-n_e$ packets to be transmitted; analogously, we service the FTP queues according to the ModCod and PF index order to select the first n_{FTP} packets to be transmitted for the FTP class.
- **Step 2**: *Selection of ModCod and number of CBR and FTP packets to be transmitted by the next frame in the presence of no urgent CBR packets*: if at the scheduling instant there are no CBR packets whose deadlines expire within the next scheduling interval, we consider for each ModCod j all the combinations of allocable CBR packets and FTP packets by using packets belonging to ModCod j or higher and we select the combination that maximizes the throughput achieved with the BBFrame transmission. We repeat this process for each ModCod and, at the end, we select the ModCod $j*$ and the related packet allocation (n_{CBR} and n_{FTP} values) that maximizes the throughput achieved with the BBFrame transmission. At this point we have determined the ModCod $j*$ selected for the transmission of the next frame, the number of allocated CBR packets, n_{CBR}, and the number of allocated FTP packets, n_{FTP}. We now determine exactly which packets have to be serviced from the FTP and CBR traffic queues. We service the CBR queues according to the ModCod and EDF order to select the first n_{CBR} packets; analogously, we service the FTP queues according to the ModCod and PF index order to select the first n_{FTP} packets.

Note that our proposed UBMT scheduler could be also suitable to manage rt-VBR flows instead of CBR ones. The scheme is exactly the same assuming that rt-VBR flows have a packet of

fixed size. If we choose to integrate the management of CBR, rt-VBR and FTP flows, a new scheduling scheme has to be defined. Further details are beyond the scope of this paper.

IV. Results and Comparisons

The performance of our UBMT scheduler has been compared with two alternative simplified scheduling approaches:

- *Urgency-Based* (UB) strategy – We select the ModCod permitting to transmit the maximum number of CBR packets; the remaining part of the BBFrame is filled in with FTP packets. This approach means to perform practically Step 1 of the UBMT scheme in every frame, also in the presence of no urgent CBR packets.
- *Maximum Throughput* (MT) strategy – We select the ModCod that maximizes the throughput achieved with the BBFrame transmission. This approach means to perform Step 2 of the UBMT scheme in every frame, even in the presence of urgent CBR packets.

An ns-2-based simulator has been built that represents the DVB-S2 forward link scheduling scenario reading the channel from a trace and supporting three techniques, that is UBMT, UB, and MT [5]. These scheduling schemes are compared in terms of video packet dropping probability, P_{drop}, due to CBR packet deadline expiration, average goodput per FTP flow, and utilization of resources by relating the total aggregated carried out traffic to the (maximum net) capacity C_s. Figure 8 shows the three performance parameters of interest in a configuration with 100 FTP flows and a varying number CBR flows. We can note that MT and UBMT attain the best utilization of resources, but UBMT allows a lower P_{drop} value. As expected, the UB technique guarantees the lowest P_{drop} value at the expenses of the worse utilization of resources since FTP traffic has the worse performance. Hence, considering the management of both CBR and FTP traffic flows, we can state that UBMT permits to achieve the best results. UBMT is a quite optimized and complex technique whose performance could be used also as a reference to benchmark other simpler techniques. Finally, Figure 9 evaluates the UBMT performance increasing the number of FTP flows (with 15 CBR flows) and considering different TCP versions, such as: *NewReno Impatient* (NRI), *NewReno Slow-but-Steady* (NRSBS), and SACK. Since channel misalignment intervals entail bursty errors, the interest is to investigate the performance of different TCP versions in the presence of bursty errors in the aim of a cross-layer study. Note that the average length of a burst of errors seen by an FTP flow depends on the scheduling strategy and the number of flows to be serviced; both aspects determine different time allocations for the service of a given traffic flow. In fact, the lower the number of FTP flows, the longer the average burst of errors experienced by a given FTP flow. In this cross-layer study, the interest of the graphs in Figure 9 is that they permit to appreciate that experienced error burstiness for low number of FTP users is better supported by SACK that allows the recovery of multiple packet losses in a window of data (for high number of FTP users, all the TCP versions have similar performance).

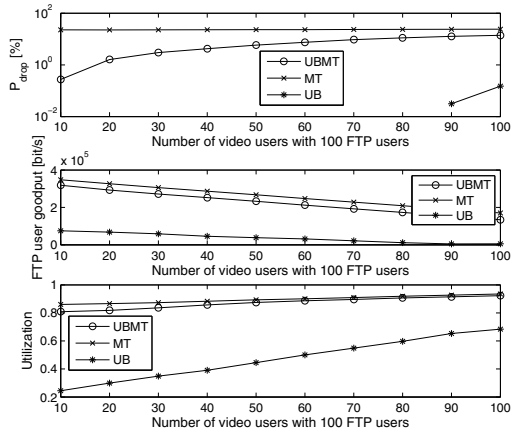

Fig. 8. Performance comparison of the scheduling strategies.

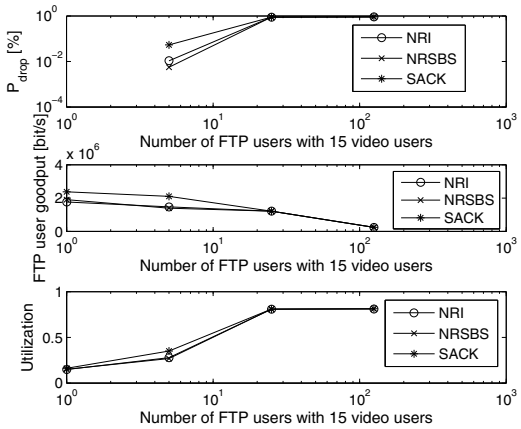

Fig. 9. UBMT performance with different TCP versions.

As a conclusion we may state that UBMT is an interesting technique, optimizing the utilization of resources for DVB-S2 with two traffic classes. A future study is needed to consider the train scenario with more critical channel conditions, to investigate simplified queue architectures for the scheduler and to account for extra delays due to higher-layer coding.

Acknowledgments

This paper has been carried out within the framework of the SatNEx II (IST No. IST-027393) FP6 network of excellence.

References

[1] ETSI, "Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and other Broadband Satellite Applications (DVB-S2)", EN 302 307.

[2] D. Breynaert, M. d'Oreye de Lantremange, "Analysis of the Bandwidth Efficiency of DVB-S2 in a Typical Data Distribution Network", *in Proc. of CCBN2005*, Beijing, March 21-23, 2005.

[3] A. Morello, V. Mignone, "DVB-S2 - Ready for Lift off", *EBU Technical Review*, October 2004.

[4] S. Scalise, H. Ernst, G. Harles, "Measurement and Modeling of the Land Mobile Satellite Channel at Ku-Band", *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 2, pp. 693-703, 2004.

[5] NS-2 Network Simulator (Vers. 2.29), URL: http://www.isi.edu/nsnam/ns/ns-build.html.