Assessing the best strategy to improve the stability of scalable video transmission in MANETs

Pedro A. Chaparro, Jesus Alcober
BAMPLA Research Group
Technical University of Catalonia
08034 Barcelona, Spain
e-mail: {pedro.alonso.chaparro, jesus.alcober}@upc.edu

Janio Monteiro
University of Algarve
Faro 8000-810, Portugal
e-mail: jmmontei@ualg.pt

Carlos T. Calafate, Juan-Carlos Cano
Pietro Manzoni
Networking Research Group GRC
Universidad Politécnica de Valencia
46022 Valencia, Spain
e-mail: {calafate, jucano, pmanzoni}@disca.upv.es

***Abstract*—Mobile Ad Hoc Networks (MANETs) have been an important research topic for the last years, playing a crucial role within the fast growing sector of mobile communications. At the same time, video applications over mobile devices are becoming widely used by nowadays mobile clients, where the quality in the transmission of such contents will determine the success of these applications in the future. Therefore, it is mandatory to find the best strategies to guarantee a good Quality of Service (QoS) to the end-user. In this work we present a set of novel strategies to improve the performance of video transmission over MANETs. These new strategies are based on distributed admission control protocols which has proved to be helpful at achieving an efficient video transmission system. Experimental results show that, when adopting the new strategies to determine the optimal number of layers to transmit, we can achieve better results compared to other existent approaches in terms of idle time periods, fairness and delay.**

***Keywords*-scalable video, distributed admission control, ad hoc networks;**

I. Introduction

Wireless communications are becoming increasingly important in our everyday lives. Among all the different kinds of available wireless networks nowadays, Mobile Ad Hoc Networks (MANETs) [1] are expected to be widely deployed in the future, and video data will represent a significant percentage of the network traffic (almost 66 percent of the world's mobile data traffic will be video by 2014 according to Cisco [2]).

In terms of video coding, the Joint Video Team (JVT) from ISO/IEC MPEG and the ITU-T VCEG developed a scalable video coding (SVC) amendment [3] for the H.264/AVC standard [4] that augments the original encoder's functionality to generate several layers of quality. Accordingly, SVC provides scalable video streams [5] which are composed of a base layer and one or more enhancement layers. Enhancement layers may enhance the temporal resolution (i.e. the frame rate), the spatial resolution (i.e. image size), or the signal-to-noise ratio resolution (i.e. SNR) of the content represented by the lower layers. By using the SVC scalability layers, video flows with different levels of quality could be transmitted efficiently over both wired and wireless networks, allowing seamless adaptation to both, the available bandwidth and the characteristics of the terminal.

Supporting real-time video on MANETs is a complex task. Therefore, new techniques must be adopted to offer acceptable QoS levels to video traffic.

DACME-SV (Distributed Admission Control for MANETs Environments - Scalable Video) [6], is a QoS framework supporting scalable video streaming in mobile ad hoc networks based on distributed admission control and video traffic awareness. It relies on a periodic probing process to measure the available bandwidth and the end-to-end delay on the path. DACME-SV adopts a cross-layer approach to determine the optimum number of video layers to transmit at any given time, avoiding network congestion and guaranteeing an acceptable video quality at the destination. Also, it promotes fairness between video flows in terms of resource consumption, and it guarantees a significant reduction of the idle times experienced by users during periods of network saturation, thus increasing the video play-out time in reception for all users. However, its criteria to decide the number of layers to transmit can be improved. In this work we go beyond in the analysis of the layer transmission criteria used by DACME-SV, proposing some novel strategies to improve DACME-SV's performance.

The rest of this paper is organized as follows: Section II offers an overview of some works related with scalable video support on mobile ad hoc networks. In Section III we describe the original DACME-SV which supports scalable video through cross-layer interactions. Section IV then presents our novel strategies to improve its performance. In Section V we present our experimental results. Finally, Section VI concludes the paper.

II. Related Works

In the past, several studies have addressed the problem of video transmission in mobile ad hoc network environments. Lindeberg et al. [7] present an analysis of the different challenges of video streaming over MANETs. Examples include cross-layer optimization, caching and replication, and packet prioritization. Cross-layer optimization typically leverages multiple description video coding and multipath routing to provide the receiver(s) with sufficient video quality. Qin and Zimmermann [8] describe the potential use of SVC in mobile networks. Furthermore, they outline use cases of mobile media delivery which can benefit from using SVC. Schierl et al. [9] present an approach that combines the benefits of cooperative interaction of clients and relay nodes in a overlay network on top of a MANET for enhancing reliability in connectivity. For a suitable application layer QoS support, they rely on Scalable Video Coding (SVC) and application layer forward error correction (AL-FEC). Fiandrotti et al. [10] introduce a traffic prioritization algorithm suitable for the transmission of both H.264/SVC and H.264/AVC video over 802.11e ad hoc wireless networks. The proposed algorithm exploits the traffic prioritization capabilities offered by 802.11e to provide better protection to the most perceptually important parts of a video while achieving efficient network resource usage.

To the best of our knowledge, there is currently no proposal in the open literature of a QoS framework for MANETs offering scalable video awareness at the admission control layer.

III. Original DACME-SV: overview

DACME-SV [6] is an extension to the original DACME framework [11] endowing it with scalable video awareness. The main goal of DACME-SV was allowing to dynamically adjust the number of layers transmitted according to end-to-end bandwidth availability.

For a joint treatment of video layers to be possible at the admission control layer, the QoS specification defined by the application includes the average bandwidth consumed by each video layer (bw_k), and the corresponding source ports to keep track of them. Upon receiving these specifications, DACME-SV sources acquire end-to-end path information in terms of bandwidth availability to decide how many video layers can be transmitted over the network without congesting it, thus guaranteeing that traffic will not exceed the network's capacity; if the path's characteristics improve, or if, on the contrary, they become worse, DACME-SV will automatically adapt to such network variability.

DACME-SV [6] relies on probing packets to test the bandwidth availability on an end-to-end path. As shown in [11], this method is highly accurate. The DACME-SV agent compares the result obtained through this process with a pre-established bandwidth threshold for each video layer. If the bandwidth available is greater than the one required for layer *n*, the agent takes a decision and will change his state to *n*. Afterward, and based on the agent's state, it will transmit all the packets belonging to video layer *n* and below. Thus, the framework allows adapting the video quality to the current resource availability. Algorithm 1 presents the DACME-SV admission control mechanism supporting multi-layer video awareness.

This algorithm works in the following manner: initially it estimates a confidence interval for the available bandwidth. If that interval contains possible data rates for the aggregated video traffic, it generates additional probes to refine the process (up to the maximum allowed) until the normalized value for the interval becomes less than δ (which in our experiments we have set to 15%). Finally, the *decision_criteria()* function proceeds to check how many consecutive video layers (starting from the base layer) can be served by the network according to its current conditions. This way, DACME-SV is able to dynamically regulate the quality of streaming video sessions by transmitting more or less video layers.

Algorithm 1 Enhanced admission control mechanism supporting multi-layer data flows.

```
Variable i = highest video layer
After receiving a bandwidth probe reply do {
    correct the bandwidth estimation $B_e$ to avoid over-estimation of resources.
  calculate 95% confidence interval for $B_e$: $[b_{min}, b_{max}]$
  If ($\exists n : \sum_{k=o}^{i} bw_k \epsilon [b_{min}, b_{max}]$ && $\frac{b_{max}-b_{min}}{B_e} > \delta$ && number of probes used < max. allowed)
      then send a new probe
  else   decision_criteria();
}
```

IV. Proposed strategies for optimal video layer selection

Although the original DACME-SV algorithm shows a good performance [6], there are still some issues that can be improved to have a better overall framework performance. In particular, the items that offer room for improvement are: (i) compliance with maximum delay requirements, (ii) fairness in terms of the quality experienced by the different users, and (iii) idle time periods, which should be reduced to a minimum to avoid user annoyance.

With the aim of improving the performance of DACME-SV we present a collection of strategies for the decision criteria function exposed above that allow extending DACME-SV's original functionality. We have them classified in two main groups: (i) static strategies, and (ii) adaptive strategies. Below we describe each of these strategies, and later we evaluate them based on network metrics like throughput, delay, network utilization, fairness, and idle times.

A. Static strategies (SS)

Static strategies merely depend on past DACME-SV states. The aim of these strategies is promoting fairness by applying smooth transitions between layers, while simultaneously decreasing idle time periods. Some of them perform a pushing process, which consists of transmitting layers for a guard period despite not enough bandwidth is available. This strategy allows mitigating bandwidth fluctuations associated with short-termed link quality disruptions. Notice that this pushing process reduces idle time periods by avoiding complete video signal blockage. To carry out this pushing process it is necessary to establish limits to avoid insisting on the transmission of a layer when the network is unable to support it.

Algorithm 2 Step-by-step layer variation.

```
While ( $b_{min} < \sum_{k=o}^{i} bw_k$ && i >= 0 )
  if( previous bandwidth flag lower by one)
      reset counters
      goto ChangeLevel
   else if( previous bandwidth flag bigger by one)
      goto ChangeLevel
   else if(counter_i < PushThreshold)
     increment counter
      goto ChangeLevel
  decrement i
 end While
ChangeLevel:set bandwidth flag to i
```

Algorithm 3 Step-by-step increase, sudden decrease.

```
While ( $b_{min} < \sum_{k=o}^{i} bw_k$ && i >= 0 )
  if( previous bandwidth flag is lower by one)
     goto ChangeLevel
  decrement i
 end While
ChangeLevel: set bandwidth flag to i
```

1) Static Strategy #1: Step-by-step layer variation: This strategy causes a smooth transition between layers, meaning that the number of layers always increases or decreases gradually (one by one), avoiding abrupt transitions. Thus, if a station has enough bandwidth to jump from layer 1 transmission to layer 5, the algorithm forces a progressive layer-by-layer increase (layers 2, 3, and 4 are selected before layer 5). Similarly, when the available bandwidth diminishes, this strategy will gradually decrease the number of layers, thus pushing layers during a short period even if the bandwidth reported is not enough to accommodate them all. The proposed strategy is illustrated in Algorithm 2.

2) Static Strategy #2: Step by step increase, sudden decrease: The second strategy is an asymmetrical solution where, similarly to static strategy #1, the number of transmitted layers is progressively incremented when the available bandwidth is enough. However, in contrast with static strategy #1, when the available bandwidth decreases the number of transmitted layers must fit the current network conditions, thus provoking a more sudden decrease. In this case there is no bandwidth pushing as in the former strategy, thus being a more conservative strategy (see Algorithm 3).

3) Static Strategy #3: Step-by-step layer increase and sudden decrease with base layer pushing: The third strategy is a combination of the two previous strategies. Now, when the available bandwidth is enough to fit a specific number of layers, DACME-SV agent will do a gradual increase, as explained before. Also, when the bandwidth is reduced due to congestion, the agent will gradually decrease the number of layers being transmitted, i.e., similarly to strategy 1. However, the basic difference is that, when the available bandwidth is not enough to fit the base layer, DACME-SV will carry out a pushing process avoiding that the transmission for that user could be completely blocked. As in the first strategy, a pushing counter is defined here for the layer we want to force. If, after some period of time, the network state does not change, and the pushing counter limit is reached, the DACME-SV agent will give up and will switch to the idle state. This means that video transmission is blocked. If later the network's state improves, DACME-SV will start transmitting and will reset the pushing counter value. This strategy is described in Algorithm 4.

Algorithm 4 Step-by-step layer increase and sudden decrease with base layer pushing.

```
 While ( $b_{min} < \sum_{k=o}^{i} bw_k$ && i >= 0 )
   if( previous bandwidth flag is lower by one)
      goto ChangeLevel
   else if(BaseLayerPushThrelshod < limit)
      goto ChangeLevel
   decrement i
  end While
ChangeLevel: set bandwidth flag to i
```

B. Adaptive strategies (AS)

Adaptive strategies depend on the network topology at a given time (e.g., number of hops towards the destination). They allow modifying the parameters used by the decision function on-the-fly, adapting DACME-SV's behavior to topology changes so that different DACME-SV video sources will have different threshold values in their decision functions. This way we can improve the fairness and reduce the idle time periods, while also improving throughput and network usage performance.

1) Adaptive Strategy #1: Making reserved bandwidth proportional to the number of hops: The original DACME-SV uses a fix reservation bandwidth value to decide whether to admit new traffic. The goal is to improve video transmission resilience by allocating some of the bandwidth to cope with link-capacity fluctuations and best-effort traffic. However, using a constant value for this parameter could be at times a waste of network resources. In our adaptive strategy #1 we propose making the reserved bandwidth value proportional to the number of hops between source and destination nodes, where the number of hops is determined by the TTL header field. This way we impose higher video admission restrictions as the number of hops increases. This strategy is described in Algorithm 5.

2) Adaptive Strategy #2: Restricting the maximum number of layers according to the number of hops: This solution, similarly to the previous one, enforces rules that make admission requirements proportional to the number of hops. The main difference is that, in this case, we introduce a new parameter to represent video transmission capacity (C) requirements, which is calculated as follows:

$$C_k = \frac{bw_k . N_{hops}}{BW_{LinkLayer}}$$

Algorithm 5 Bandwidth reserved proportional to number of hops.

```
$N_{hops}$=number of hops
$bw_k = (0.2 * N_{hops} * bw_k) + bw_k$
 While ( $b_{min} < \sum_{k=o}^{i} bw_k$ && i >= 0 )
   decrement i
  end While
ChangeLevel: set bandwidth flag to i
```

Algorithm 6 Restricted maximum number of layers according to number of hops.

```
 While ( $C < \sum_{k=o}^{i} C_k$ && i >= 0 )
   decrement i
  end While
ChangeLevel: set bandwidth flag to i
```

where bw_k is the bandwidth required by each video layer, N_{hops} is the number of hops between the video source node and the destination node, and $BW_{LinkLayer}$ is the maximum throughput achievable at the link layer. In our case we target 801.11g, and so this value is of about 27 Mbps; the capacity required by each layer is defined by C_k. Algorithm 6 illustrates the strategy adopted.

V. DACME-SV Strategies Performance Evaluation

In this section we analyze the performance of the proposed strategies using a simulated MANET environment. The simulation platform used is ns-2 [12]. To perform our evaluation we setup a 870x870 meters scenario with 50 nodes moving according to the random way-point mobility model at a speed of 5 m/s. All nodes are equipped with an IEEE 802.11g interface transmitting at 54Mbit/s; the transmission range is 250 m. Additionally, all the nodes offer MAC level QoS according to the IEEE 802.11e standard. At the network layer, we employ AODV [13] as the routing protocol. Simulation time is set to 420 seconds, but we start video transmission after 100 seconds to make the measurements on a steady-state environment. All measurements were made over a period of 300 seconds and averaged over ten simulation runs, varying the number of video source-destination pairs from 1 to 10 for each of the strategies proposed. With respect to video simulation settings, we used real video traffic by adapting the video encoder output to trace formats compatible with ns-2. The video encoder used was the scalable extension to H.264 (JSVM), version 9.10.

The video sequence used was the well known CREW sequence [14], repeated during 300 seconds. The sequence was captured at 30 frames per second and encoded with a GOP size of 16 at three different resolutions - QCIF, CIF and 4CIF - with two SNR levels for each resolution. Thus, we have a total of six video layers, and the output bit rate varies from 170 kbps (layer 0 alone) to 2.2 Mbps (layers 0 to 5), depending of the number of layers transmitted (see Table I). The number of layers and their characteristics were selected taking as reference a previous work [15].

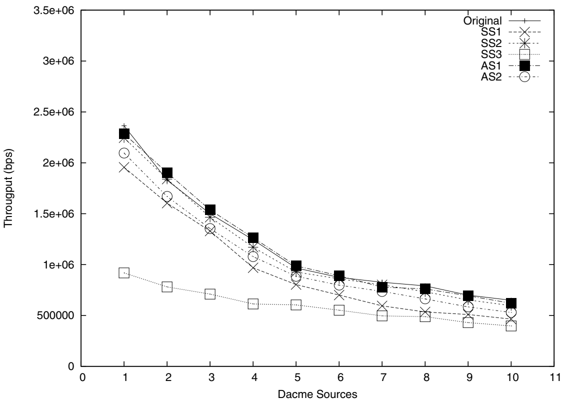

Figure 1. Throughput results for DACME-SV and the different strategies when increasing the number of video sources.

In our experiments we also introduce four FTP/TCP sources as background traffic in the best-effort category of IEEE 802.11e to make the MANET operate under more realistic assumptions. Concerning video traffic, DACME-SV assigns all packets to the Video category of IEEE 802.11e (AC_VI). For comparison we select the original DACME-SV approach.

A. Throughput, delay and channel utilization results

We first focus on the experimental results obtained in terms of throughput, delay and channel utilization. Notice that our main goals with DACME-SV were to reduce the idle times of real-time sessions, to increase the fairness among competing users and to improve channel utilization. The new decision strategies aim for the same goals, while improving the performance of the original DACME-SV. In terms of average throughput and delay values, these should be maintained similar to those for the standard DACME implementation.

Figure 1 shows the results obtained in terms of throughput. When there is more than a video source active, the throughput measured corresponds to the average value among them all; the same applies to delay and channel utilization results. Overall, we observe that the throughput values are quite similar in all the strategies, showing the adaptive solutions better results than the static ones.

Concerning delay, figure 2 shows that most of the strategies proposed improved the delay values in comparison with the original DACME-SV. To better understand this delay decrease, Figure 3 shows the average network

Table I
Characterization of the different video layers involved in terms of bit rate, spatial resolution and quality.

Layer	Aggregated Bit rate (kbps)	Resolution	SNR
0	178.5	QCIF	Low
1	190.2	QCIF	High
2	626.5	CIF	Low
3	676.8	CIF	High
4	1959.7	4CIF	Low
5	2236.0	4CIF	High

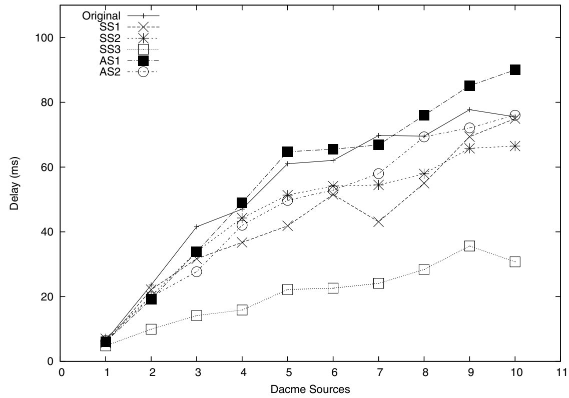

Figure 2. Delay results for DACME-SV and the different strategies when increasing the number of video sources.

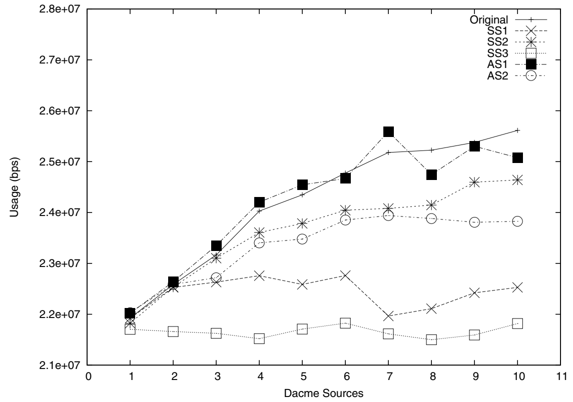

Figure 3. Network utilization results for DACME-SV and the different strategies when increasing the number of video sources.

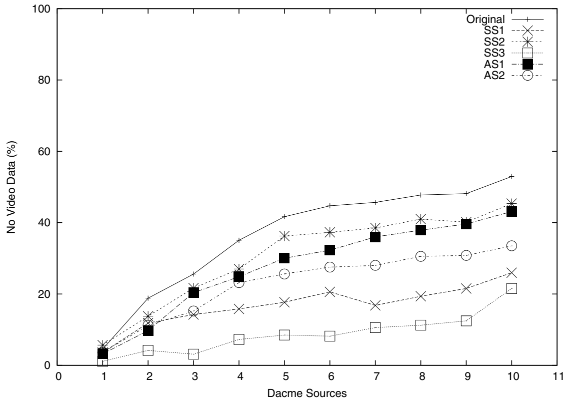

Figure 4. Percentage of idle time improvements for DACME-SV with respect to the original DACME when increasing the number of video sources.

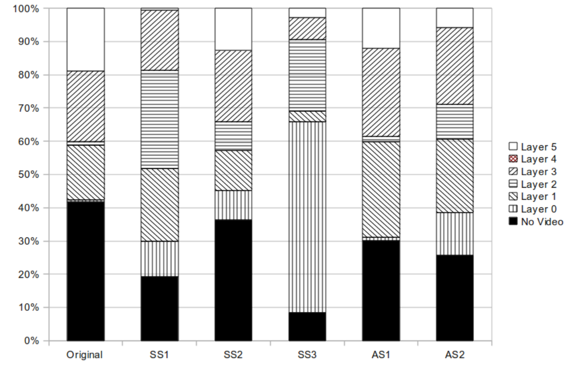

Figure 5. Comparison of the amount of video layers decoded when using different strategies.

utilization when varying the number of video sources. We can see that less data is transmitted in the network since most strategies are more restrictive than the original DACME-SV. Thus, in the presence of source-destination pairs separated by a high hop distance, their chances of transmitting at high data rates decreases.

B. Performance in terms of Idle times

In this section we present experimental results related to idle times. We define *idle time* as any period of time during which communication is interrupted due to lack of resources, being impossible to transmit any video data during such period. Obviously, too many idle time periods will provoke the end-user experience to be poor. Hence, our goal is to reduce the frequency of these interruptions.

Figure 4 shows the percentage of idle times achieved by each of the new DACME-SV decision algorithms compared with the original version. All the new strategies show improvements in this metric, clearly reducing idle times. However, the results obtained show that static strategies enforcing layer pushing (SS1 and SS3) perform better than the rest. For instance, in the case of five active video DACME-SV sources, static strategy #3 reduced idle times by 79.83% compared to the original DACME-SV.

To better understand how the new strategies are able to reduce idle times without consuming a greater amount of bandwidth, we now detail the degree of video quality perceived by users, which is strictly related to the number of H.264/SVC video layers available for decoding.

Figure 5 presents a comparison between all the strategies in terms of video visualization experience, as perceived by the end user, for the five video sources scenario. In black we represent the percentage of time during which video is not transmitted because the network is unable to support it (due to lack of bandwidth). Notice that the percentage of idle time is decreased in all of the strategies in comparison to the original DACME-SV. This is associated with a fairer video transmission; in particular, observe how static strategies #1 and #3 are able to communicate 83% and 96% of the time, respectively. This is due to the pushing process used, as mentioned before. In this particular example no results for *Layer 4* appear because its bandwidth requirements are very similar to those of layer 5 (High video level), meaning that, when the available bandwidth in our experiments was high, it was always large enough to accommodate all video layers.

C. Measuring resource usage fairness

To complete our evaluation we now concentrate on the issue of fairness. Our purpose is to assess if users benefit from a similar resource usage or if, on the contrary,

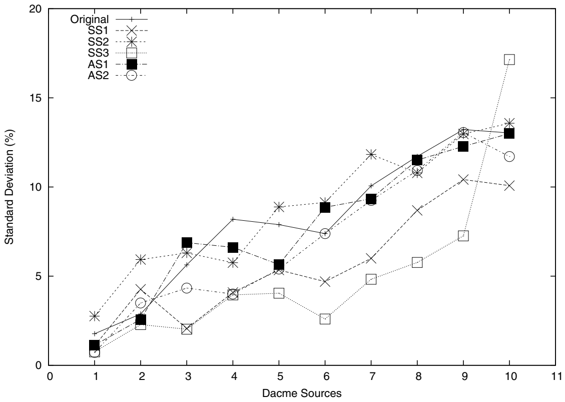

Figure 6. Standard deviation results for idle times for all the strategies when increasing the number of video sources.

some users experience excellent conditions while others experience very poor ones.

The metric we have chosen was the normalized standard deviation for idle times. Notice that lower values indicate a similar experience by all users, while high values indicate that some users are experiencing fluid communications conditions, while others are often experiencing communication disruptions.

Fairness results are presented in Figure 6, where we can observe that not all the static strategies using the pushing process show the best results, achieving reductions of up to 45.73 percent as the case of static strategy #3 for five video source nodes. Adaptive strategies improve fairness compared to the original DACME-SV.

Overall, we consider that both groups of strategies proposed, static and adaptive ones, have successfully met the design goals set, each offering its own set of trade-offs. Overall, we find that, indeed, the greater flexibility offered by the H.264/SVC codec was translated into network-level and user-level improvements.

VI. Conclusions

In this paper we presented a set of enhancements to DACME-SV grouped in static and adaptive strategies. We showed the benefits of using each of these improvements and analyzed them in terms of network performance.

Results showed that the design goals for both static and adaptive strategies was achieved. In particular, we improved the performance of the original DACME-SV implementation by decreasing the frequency of interruptions on video communications, while simultaneously improving fairness among users by favoring similar conditions for QoS flows.

VII. Acknowledgments

This work was partially supported by the *Ministerio de Educación y Ciencia*, Spain, under Grants TIN2008-06441-C02-01 and TSI2007-66637-C02-01.

References

[1] I. E. T. Force, "MANET Working Group Charter." http://www.ietf.org/html.charters/manet-charter.html.

[2] "Cisco visual networking index: Forecast and methodology, 2009-2014." Technical report, Cisco Systems, June 2010.

[3] ITU-T and ISO/IEC JTC1, JVT-W201, "Joint Draft 10 of SVC Amendment," *Joint Video Team (JVT) of ISO-IEC MPEG*, April 2007.

[4] Snjezana Rimac-Drlje, Ognjen Nemcic, and Mario Vranjes, "Advance Video Coding for Generic Audiovisual Services," *ITU-T Rec. H.264|ISO/IEC IS 14496-10 AVC*, vol. 3, September 2005.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1103–1120, September 2007.

[6] P. Chaparro, J. Alcober, J. Monteiro, C. Calafate, J.-C. Cano, and P. Manzoni, "Supporting scalable video transmission in manets through distributed admission control mechanisms," in *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, pp. 238 –245, 17-19 2010.

[7] M. Lindeberg, S. Kristiansen, T. Plagemann, and V. Goebel, "Challenges and techniques for video streaming over mobile ad hoc networks," *Multimedia Systems*, pp. 1–32, 2010. 10.1007/s00530-010-0187-8.

[8] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile video transmission using scalable video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, pp. 1204 –1217, sep. 2007.

[9] T. Schierl, S. Johansen, C. Hellge, T. Stockhammer, and T. Wiegand, "Distributed rate-distortion optimization for rateless coded scalable video in mobile ad hoc networks," vol. 6, pp. VI –497 –VI –500, sep. 2007.

[10] A. Fiandrotti, D. Gallucci, E. Masala, and E. Magli, "Traffic prioritization of h.264/svc video over 802.11e ad hoc wireless networks," pp. 1 –5, aug. 2008.

[11] C. Calafate, M. Malumbres, J. Oliver, J. Cano, and P. Manzoni, "Qos support in manets: a modular architecture based on the ieee 802.11e technology," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, pp. 678 –692, May 2009.

[12] K. Fall and K. Varadhan, *ns notes and documents. The VINT Project*. UC Berkeley, LBL, USC/ISI and Xerox PARC, February 2000. Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.

[13] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc Distance Vector (AODV) Routing," *RFC3561 , IETF*, July 2003.

[14] "H.264/SVC Test Sequences (2008, July 27). Institute of Signal Processing, Hannover University, Germany." [Online]. Available: ftp://ftp.tnt.uni-hannover.de/pub/svc/.

[15] Janio M. Monteiro, Carlos T. Calafate, and Mario S. Nunes, "Evaluation of the H.264 Scalable Video Coding in Error Prone IP Networks," *IEEE Transactions on Broadcasting*, vol. 54, pp. 652–659, September 2008.