Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 31 (2008) 1290–1300
computer communications
www.elsevier.com/locate/comcom

A traffic aggregation and differentiation scheme for enhanced QoS in IEEE 802.11-based Wireless Mesh Networks

R. Riggio [a,b,*], D. Miorandi [b], F. De Pellegrini [b], F. Granelli [c], I. Chlamtac [b]

[a] Department of Electrical and Computer Engineering, University of Florida, 435 Engineering Building, P.O. Box 116130, Gainesville, FL 32611, USA
[b] CREATE-NET, Via alla Cascata 56/C, 38100 Trento, Italy
[c] Department of Communication and Information Technologies, University of Trento, Via Sommarive 14, Trento, Italy

Available online 1 February 2008

Abstract

Wireless Mesh Networks are currently emerging as a promising paradigm for broadband ubiquitous Internet access. However, their distributed nature raises many challenges when facing the increasing demand for multimedia applications, which require a tight control over the system's available resources. In this paper, we address such issue by introducing a mechanism combining service differentiation and packet aggregation in IEEE 802.11-based WMNs. Our architecture does not require any modification to the IEEE 802.11 MAC and can be readily deployed exploiting off-the-shelf hardware. The proposed solution has been implemented as an extension to the MIT Roofnet platform. Measurements run over a WiFi testbed show a large gain in the voice capacity attained. The source code, released under the BSD License, is made available to the research community.

Keywords: Wireless Mesh Networks; IEEE 802.11; Traffic aggregation; Service differentiation; Experimental measurements; VoIP; Roofnet

1. Introduction

The provisioning of Quality of Service (QoS) can be considered a mandatory requirement for any telecommunication system able to support multimedia services. However, the current Internet lacks a widely deployed framework for supporting QoS. One of the reasons is that QoS mechanisms are mostly needed when network resources are scarce and real world experience has proved that is often cheaper to upgrade to higher capacity links or equipments (over-provisioning) than to deploy Internet-wide QoS solutions. On the other hand, the current bottleneck is represented by the last mile of the Internet connection. Therefore, techniques able to provide QoS over access networks are believed to represent a viable solution to enhance the accessibility to multimedia services.

Wireless LAN (WLANs) are currently setting themselves as *the* standard access network technology, due to the widespread availability of extremely cheap hardware and the deployment of a huge number of hot-spots. Nonetheless, WLAN technology still suffers from the need of wiring each access point through fiber of high-speed DSL links. In such a scenario, Wireless Mesh Networks (WMNs) gained considerable attention over the last few years, emerging as a promising paradigm for broadband ubiquitous Internet access [1]. As opposed to WLANS, WMNs exploit a multi-hop wireless back-haul in order to deliver Internet connectivity to the end-users. Albeit WMNs could interface, through suitable gateways, networks based on different radio technologies (3G, WiMAX, WiFi, Bluetooth), in this paper we will focus our attention on IEEE 802.11-based WMNs.

The distributed and decentralized nature of WMNs presents many challenges when facing the increasing demand

* Corresponding author. Address: CREATE-NET, Via alla Cascata 56/C, 38100 Trento, Italy. Tel.: +39 0461 314 960; fax: +39 0461 314 972.
E-mail addresses: roberto.riggio@create-net.org (R. Riggio), daniele.miorandi@create-net.org (D. Miorandi), fdepellegrini@create-net.org (F. De Pellegrini), granelli@dit.unitn.it (F. Granelli), imrich.chlamtac@create-net.org (I. Chlamtac).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.01.037

for multimedia applications, which require a tight control over the system's available resources. Furthermore, as proved in recent studies [2], WMNs scalability problems pose additional constraints so that ensuring the required QoS parameters appears a challenging task even for a small number of hops (2–3). But, despite this is considered a strategic goal to achieve, little efforts have been dedicated to investigate efficient techniques for supporting QoS in WMNs. The vendors which have been selling wireless mesh solutions of course do implement some form of QoS policies, but they are obviously very reluctant to release those informations. Hence the research in WMNs field lacks from a comprehensive QoS perspective.

In this work, we aim at enhancing the perceived quality of experience combining service differentiation and packet aggregation in IEEE 802.11-based WMNs. The proposed solution is based on the use, at each node, of four priority queues and an appropriate scheduler. Each queue implements a novel packet aggregation technique which allows us to reduce service time at the MAC layer. Such an aggregation is performed on top of the MAC layer, allowing us to reduce the overhead due to both protocol headers and the contention mechanism regulating the IEEE 802.11 standard. As concerns the aggregation scheme, the novelty of the proposed approach is represented by the adaptive aggregation scheme that leverages the channel probing capabilities of mesh routers: such information is exploited in order to compute the optimal saturation burst length. To the best of the authors' knowledge, ours is the first attempt to combine routing and packet aggregation in WMNs. We tested our scheme over a IEEE 802.11-based WMN deployment exploiting both VoIP flows and saturated TCP connections. Results show a very large gain in the voice capacity attained. Notice also that our architecture does not require any modification to the IEEE 802.11 MAC and can be readily implemented using off-the-shelf hardware. The proposed solution has been implemented as an extension to the MIT Roofnet platform [2]. The code is made available to the research community being released under the BSD License.[1]

The remainder of the paper is organized as follows. Section 2 presents and discusses some related works. In Section 3, we describe the architecture of the proposed scheme. Section 4 introduces the testbed setup e describes the evaluation methodology. The architectural details and the outcomes of the measurement campaign are reported in Sections 5 and 6. Finally, Section 7 concludes the paper pointing out directions for future research.

2. Related work

The literature on the capacity of multi-hop wireless networks is extensive. In [3], the authors show how network scalability is preserved in static ad hoc networks only when the average distance between source and destination remains small as the network grows. Vice-versa non-local traffic severely impair per-node network capacity. Several techniques have been developed in order to boost network performances. In [4], multiple radio are exploited, while in [5,6] novel routing metrics capable of selecting the best route are introduced. In this paper, we address the above mentioned issues in IEEE 802.11-based WMNs by combining a traffic differentiation policy with a novel aggregation scheme capable of reducing MAC service time by compounding several frame into a single burst.

The need for service differentiation in WLANs drove the 802.11 Working Group to ratify an amendment dedicated to QoS provisioning. IEEE 802.11e introduces a new coordination function called the Hybrid Coordination Function (HCF). Within the HCF there are two access mechanisms: a non-contended polling channel access scheme *HCF Controlled Channel Access (HCCA)* and a prioritized contention scheme *Enhanced Distributed Channel Access (EDCA)*. Both EDCA and HCCA define Traffic Classes (TC). For example, best effort TCP traffic could be assigned to a low priority class, and VoIP traffic could be assigned to a high priority class. With EDCA, high priority traffic has a higher chance of being sent than low priority traffic: a station with high priority traffic waits a little less before it sends its packet, on average, than a station with low priority traffic. The HCCA way of working resembles IEEE 802.11's PCF, however, since HCCA support is not mandatory currently available APs do not support it.

We differentiate from such an approach by proposing a non-invasive solution enabling traffic prioritization at level 2.5 through the use at each node, of four priority queues and a suitable scheduler. Being implemented at level 2.5, our traffic differentiation architecture is backward compatible with the IEEE 802.11 devices that do not support the "e" QoS extension and, in principle, could fit even non IEEE 802.11 MAC protocols, resulting therefore in enhanced flexibility compared to solutions depending on MAC layer modifications.

There exists a vast literature on the modeling and analysis of the IEEE 802.11 CSMA/CA protocol. In [7], a two-dimensional Markov chain is exploited in order to model the exponential back-off algorithm of the IEEE 802.11 Distributed Coordination Function (DCF) under saturation. Among several works elaborating on the initial model, such analysis is extended to error-prone channels in [8]. The authors conclude that, for a given bit error rate, there exists an optimal packet size that maximizes the goodput. Related works on VoIP over WLANs [9–11] and UWB networks [12] proposed the introduction of a packet aggregation scheme. The technique trades off service time for packet length: the increase of CSMA/CA service time is mitigated by assembling multiple upper layer packets into a single MAC burst. Performance measures proved that the proposed MAC can significantly improve both throughput and delay performance in CSMA/CA based networks. In [13], the authors propose several performance

[1] http://www.wing-project.org.

optimizations aimed at improving the VoIP support in WMNs. Voice packet aggregation and header compression are then exploited to improve the network capacity in terms of number of voice calls supported. Extensive simulation and experiments run over a real testbed are used in order to validate the proposed approach. In [14], an analytical model is developed in order to study the impact of packet aggregation on delay. Results show that packet aggregation can significantly improve the performance of the CSMA/CA protocol. Such a result is exploited by the authors in order to provide a novel packet aggregation policy capable of optimizing both the network throughput and delay.

Our work extends both [7,12,14] by introducing a novel adaptive traffic aggregation policy capable of matching the parameters available in a real-world WMNs, namely link status and channel utilization ratio, with the requirements of an adaptive packet aggregation scheme. A closed formula allowing run-time computation of the optimal burst length based on measurable routing metrics and the number of stations in radio range is also proposed. The proposed technique does not require any modification to the IEEE 802.11 MAC and can be readily implemented over existing hardware.

3. Architectural overview

Our traffic prioritization scheme exploits the DSCP field of the IP header in order to classify incoming flows (see Section 5 for more details). Its architecture is sketched in Fig. 1. Incoming packets are classified by the *IP Classifier* and then queued into a suitable *Aggregation Buffer*. Traffic differentiation is provided by means of a *WRR Scheduler*

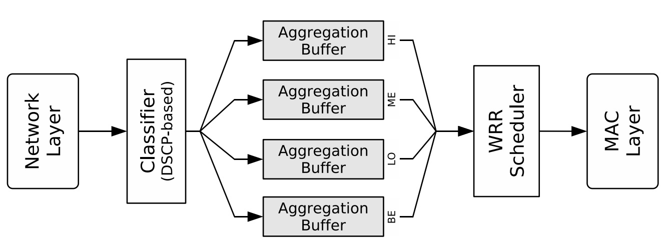


Fig. 1. Architecture of the traffic differentiation scheme supporting four different traffic classes.

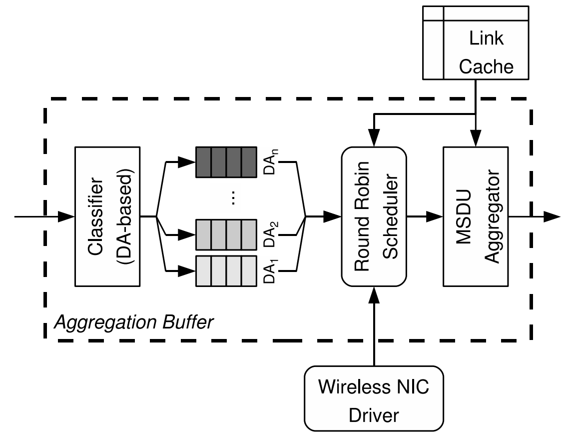


Fig. 3. Block diagram for the packet aggregator.

[15] which pulls packets from buffers, according to some input weights (see Table 2).

Each *Aggregation Buffer* concatenates several MAC Service Data Units (MSDUs) to form the data payload of a large MAC Protocol Data Unit (MPDU). The PHY header and the MAC header together with the Frame Check Sequence (FCS) are then appended in order to build the Physical Service Data Unit (PSDU). The frame format for an Aggregated MSDU (A-MSDU) is sketched in Fig. 2.

The building blocks of the *Aggregation Buffer* and their relationships are sketched in Fig. 3. Incoming MAC frames are first classified according to their destination address and then fed to a different queue. Each *Aggregation Buffer* maintains a pool of unused queues and an hash table that associates the MAC destination addresses with the corresponding queue. Unused queues are moved from the hash table to the pool, this is done in order to alleviate the need for repeated memory allocation as neighbors come and go. For each queue, an A-MSDU is generated when either an aggregation timer is expired or a burst of optimal length can be generated.

4. Evaluation methodology

4.1. Testbed configuration

The testbed exploited during our experimentation consists of five wireless mesh routers deployed in a typical

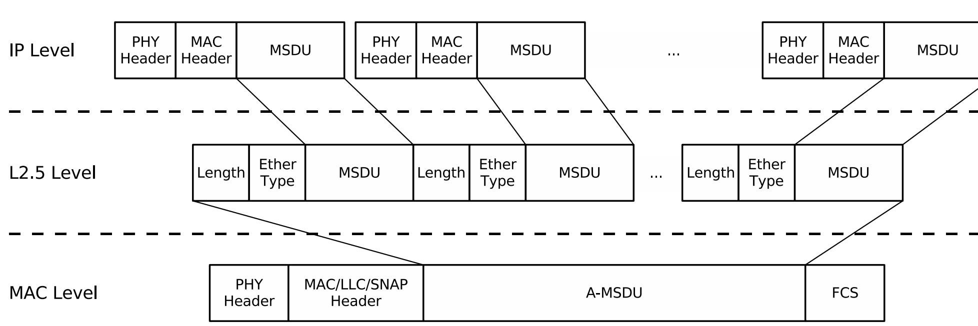


Fig. 2. Aggregated MSDU (A-MSDU) frame format.

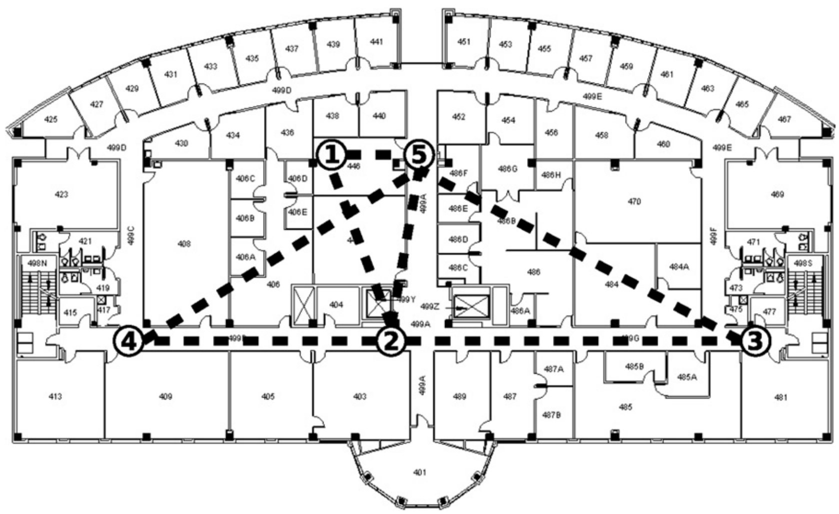


Fig. 4. Logical network topology.

office environment implementing a flat WMNs suitable to be exploited as back-haul in dual-tier architectures. Testbed's nodes are all Fujitsu notebooks model P7010D equipped with a 1.20 GHz Intel Pentium M processor and 512 MB of memory. All nodes run Debian GNU/Linux with kernel 2.6. Each node has a single IEEE 802.11b/g wireless NIC. All measurement are run with the IEEE 802.11 interfaces operating in "b" mode with RTS/CTS disabled. Network topology is sketched in Fig. 4.

Our testbed is implemented on top of Roofnet [2], an experimental IEEE 802.11-based WMN deployed at Cambridge, Massachusetts (USA) by the Massachusetts Institute of Technology (MIT). Roofnet routes packets using a modified version of DSR [16] called SrcRR [2] exploiting ETX [5] as routing metric. Routing is implemented using the Click modular router [17], developed at MIT.

A Click router is built by assembling several packet processing modules, called *elements*, forming a directed graph. Each *element* is in charge of a specific function such as packet classification, queuing, or interfacing with networking devices. Click comes with an extensive library of elements supporting various types of packet manipulations. Such a library enables easy router configuration by simply choosing the elements to be used and the connections among them. We extended the default Roofnet configuration by implementing the additional elements responsible for packet aggregation and scheduling.

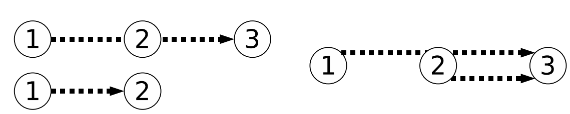
Traffic is generated using the Jugi's Traffic Generator (JTG), a freely available synthetic traffic generator [18]. Traffic is then collected at the receiver side where suitable tools are available for analysis.

4.2. Traffic differentiation

In order to validate our traffic differentiation scheme three application scenarios have been considered. The first scenario (data-set A) aims at verifying the capability of our scheme to differentiate multiple saturated TCP flows. During this set of measurements each traffic class has been fed with persistent TCP flows. The second scenario (data-set B) assesses the capability of our scheme to protect higher priority TCP flows against lower-priority UDP streams. During this set of measurements the high priority and the best effort queues are fed with, respectively, a persistent TCP flow and an UDP stream at increasing bit-rates, while the other queue are left empty. Table 1 summarizes the parameters of the UDP interfering traffic. The third scenario (data-set C) aims at assessing the capability of our scheme to differentiate multiple saturated TCP flows sharing a common intermediate hop. This scenario shares the same traffic patterns of data-set A. Fig. 5 depicts the network topologies exploited during our measurements campaign. A single and a 2-hops string topologies have been used for data-set A and B (see Fig. 5a), while data-set C utilizes a dual string topology (see Fig. 5b) involving both 1-hop and 2-hops routes.

Table 1
Parameters of the UDP noise traffic

Run	Payload length (byte)	Packet inter-generation time (ms)	Resulting bit-rate (kbit/s)
1	1460	48	≈250
2	1460	24	≈500
3	1460	12	≈1000
4	1460	6	≈2000
5	1460	3	≈4000



(a) Data-set A and B (b) Data-set C.

Fig. 5. Reference network topologies for the considered scenarios.

4.3. Packet aggregation

In order to validate our adaptive packet aggregation policy we focused our attention on VoIP applications. Being very demanding in terms of loss and delay constraints, VoIP services are traditionally an interesting benchmark case, especially when dealing with mesh structures, where multi-hop communication could introduce unpredictable delays. A typical VoIP source tends to transmit a large number of packets with a small payload, and such a combination is known to lead to large protocol overheads [13]. We have emulated each VoIP call with a single UDP stream modelled according to the parameters of the G.729.3 codec [19], a worldwide used speech codec for VoIP applications. The G.729.3 VoIP codec generates 33 pkt/s. Each packet contains three voice samples (10 bytes each) producing a final bit-rate of 8 kbit/s.

We resort to the E-Model [20] as an objective method to evaluate speech quality in VoIP systems. The outcome of an E-Model evaluation is called R-factor (R). The R-factor is a numerical measure of voice quality, ranging from 0 to 100, with 70 as lower bound for a VoIP call of acceptable quality:

$$R = 94.2 - I_{\mathrm{d}} - I_{\mathrm{ef}}, \tag{1}$$

where I_{d} is the impairment factor caused by the end-to-end delay and I_{ef} is the equipment impairment factor:

$$\begin{aligned} I_{\mathrm{d}} &= 0.024\, T_a + 0.11(T_a - 177.3)H(T_a - 177.3) \\ I_{\mathrm{ef}} &= I_{e_{\mathrm{opt}}} + C_1 \ln(1 + C_2 P). \end{aligned} \tag{2}$$

In (2), T_a and P are, respectively, the one-way delay in ms and the loss rate, $H(x)$ is the step function, and $I_{e_{\mathrm{opt}}}, C_1, C_3$ are codec-specific parameters. Their values for G.729 codec under random packet losses are, respectively, 11, 40 and 10.

In order to highlight the impact of our aggregation scheme the test performed during our measurement campaign refers to downlink traffic only. In our settings, each mesh node sustains the same traffic pattern, consisting in an increasing number of VoIP sessions generated at node number 1 and mapped as high priority traffic. In order to collect reliable measures of delays, before each experiment we synchronized each node with a common reference (node number 1) using NTP [21].

5. Service differentiation in WMNs

5.1. Differentiated services

Overall, the architecture proposed in this work does not provide deterministic QoS guarantees, but supports the differentiation of prioritized flows. As a matter of fact, providing strict performance bounds in IEEE 802.11-based WLANs is an open issue, mostly due to contention-based MAC and to the error-prone nature of the wireless medium. Notice, though, that our scheme is independent of the MAC implementation and is compatible with technologies differentiating traffic classes at the MAC level, such as, for example, the IEEE 802.11e [22] specification.

Table 2
Traffic classes supported by our architecture

Priority	DSCP	PHB	Weights (φ_i)
Best-effort (BE)	0	Default	1
Low (LO)	10	AF11	2
Medium (ME)	18	AF21	4
High (HI)	26	AF31	8

Albeit our architecture currently support four transmission priorities (see Table 2), its design allows the definition of an arbitrary number of traffic classes. No pre-emptive priority classes have been implemented, as this might have weakened our system. Non-TCP friendly flows under pre-emptive priority can make the whole system collapse, preventing also the possibility to remotely restore basic functionality. Weighted Round Robin (WRR) is exploited as scheduling policy.

The DiffServ architecture [23] provides a framework allowing classification and differentiated treatment by recommending a standardized set of traffic classes. Routers supporting DiffServ implement the so called Per-Hop Behaviors (PHBs), which define the packet forwarding properties associated with a class of traffic. The Per-Hop Behavior is indicated by encoding a 6-bit value, called the Differentiated Services Code Point (DSCP), into the 8-bit Differentiated Services (DS) field (former Type of Service, ToS field) of the IP packet header. DiffServ recommends the following PHB behaviours:

- *Default*. Typically best-effort traffic.
- *Expedited Forwarding (EF)*. Low-loss, low-latency traffic class.
- *Assured Forwarding (AF)*. It defines four separate classes. Within each class, packets are given a drop precedence (high, medium or low). The combination of classes and drop precedence yields 12 separate DSCP codes from AF11 through AF43.
- *Class Selector*. Defined to maintain backward compatibility with the IP Precedence field.

5.2. Scheduling policy

Several approaches can be exploited in order to implement a WRR scheduling policy. Our implementation deterministically builds a scheduling list according to some input weights. Such an approach is known as WRR with Weighted Fair Queueing (WFQ) spread [24]. Under the assumption that every ith flow is backlogged, such algorithm perfectly approximates the WFQ scheduling allowing us to reduce both delay variations and the presence of bursts of slots allocated to the same flow. This is done preserving the low algorithmic complexity, which is crucial in an environment where heterogeneous devices, with power

Table 3
Outcomes of the experiments involving data-set A (1-hop/2-hops)

	BE	LO	ME	HI	Total
Throughput (β_i)	357/162	719/331	1449/667	2908/1351	5433/2510
Confidence interval ($z_{95\%}$)	11.11/9.24	1.99/3.14	3.72/5.13	7.28/4.34	–
Relative ratio (ρ_i)	0.99/0.97	1.98/1.98	4.00/3.99	8.03/8.07	15

consumption and computational capacity limitations, are employed.

5.3. Evaluation

The outcomes of the experiments involving the data-set A are summarized in Table 3 for each of the four traffic classes. The first row contains the average measured throughput in kbit/s (β_i). The second row reports the confidence intervals. The third row shows the relative throughput ratios (ρ_i) for the four classes. Being φ_i the input weights, it stands:

$$\rho_i = \frac{\beta_i \sum_i \varphi_i}{\sum_i \beta_i}. \quad (3)$$

In an ideal case $\rho_i = \varphi_i$. The numerical results contained in the table reveal that, in both the 1-hop and the 2-hops scenarios, relative flows priorities are maintained. We will now analyze the results of the measurements performed using data-set B. This test aims at investigating the performances of an high priority TCP flows competing with an UDP stream at increasing bit-rates aimed at modelling background traffic. Fig. 6a shows that the introduction of our traffic differentiation scheme effectively protects the high priority flows, while, as expected, in a standard deployment the TCP performances progressively decrease when the UDP bit-rate increases. Increasing the number of hops leads to a more complex scenario. As we can see from Fig. 6b, using a standard setup, the UDP stream progressively saturate the entire bandwidth annihilating the TCP flow. However, exploiting our traffic differentiation scheme does not lead to the ideal behaviour seen in the 1-hop scenario. Instead, increasing the bit-rate of the UDP stream results in a similar share of the channel bandwidth among the two competing flows.

We found that the reason behind such a behavior lies in the scheduling policy for TCP ACKs at intermediates hops. In fact, IP packets carrying TCP ACKs are not marked as high priority traffic and are then buffered with the UDP streams, as shown in Fig. 7. They are therefore likely to be lost or to experience high latency as a result of the UDP streams non-responsiveness to link congestion.

Such behavior is most visible in data-set C where multiple full-rate TCP flows share resources at an intermediate hop (node 2). As shown in Fig. 8a the performance of the high priority flow relayed by node number 2 are drastically reduced when another high priority flow is generated locally. Please note that the 2-hops flow is generated at node number 1 at $t = 0\,s$ and ends at $t = 240\,s$, while the single hop flow is generated at node 2 at $t = 120\,s$ and ends at $t = 360\,s$. In order to mitigate this phenomenon we implemented a Wireless ACK Re-scheduling Policy (WARP) designed to give to TCP ACKs full preemptive access to the medium. Experimental results show that WARP is capable of providing a considerable performance boost to the relayed flow as can be seen from Fig. 8b. It is

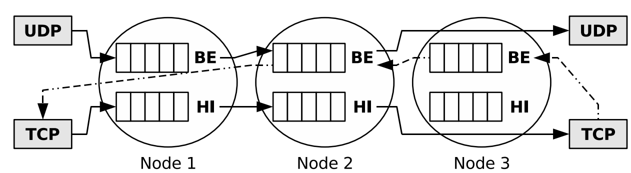


Fig. 7. Being not marked as high priority traffic, IP packets carrying TCP ACKs are buffered with the low priority UDP packets at intermediate hops.

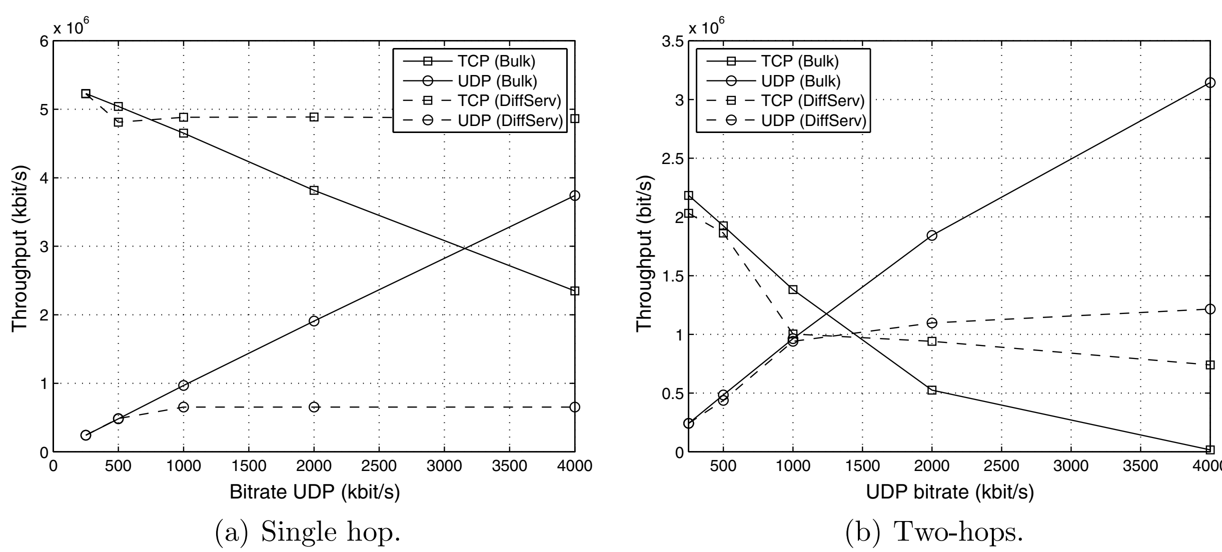
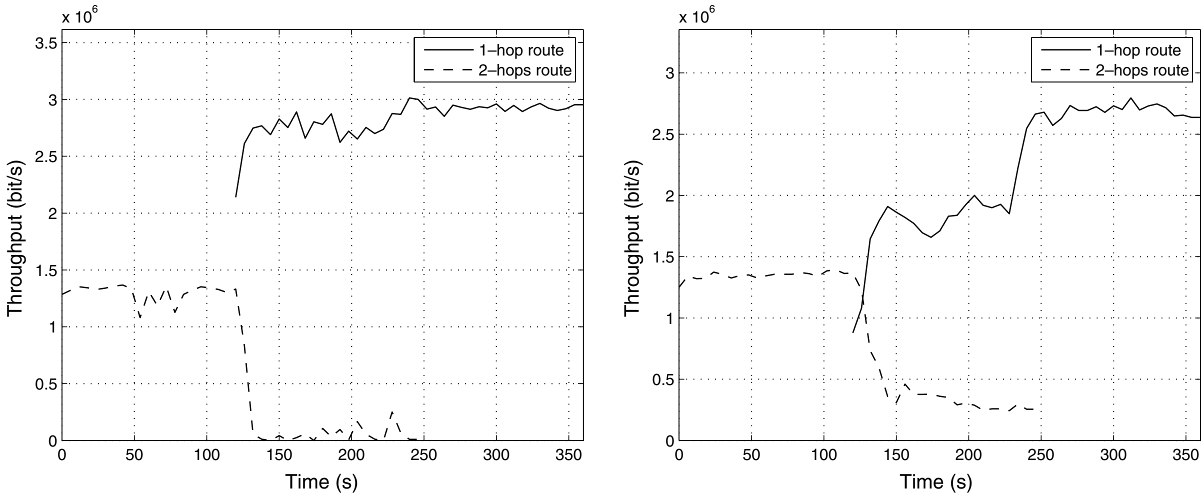


Fig. 6. Performances of a high priority TCP flows competing with an UDP stream at increasing bit-rates.



(a) Without WARP. (b) With WARP.

Fig. 8. Performances of two high priority TCP flows sharing resources at an intermediate node.

worth stressing that our approach is totally state-less and requires no admission control making it particularly suitable for resource limited mesh routers.

6. Metric oriented packet aggregation

6.1. Motivation

The overhead associated with IP packets transmitted over an IEEE 802.11 is due both to both the additional headers and to the exponential back-off algorithm required by the contention procedure defined in the standard [25]. Both types of overhead are particularly relevant in the case of small sized packets, such those used in VoIP. Fig. 9 sketches the frame structure for a IEEE 802.11 frame, as we can see the most relevant overhead is introduced by the Link Layer and by the Physical Layer headers.

The IEEE 802.11 standard [25] currently defines six modulation techniques, with data rate ranging from 1 to 54 Mbps. In order to allow the IEEE 802.11 MAC to operate with minimum dependence on the physical sublayer, the Physical Layer Convergence Procedure (PLCP) is introduced. Both PLCP Preamble and Header are transmitted at the basic rate in that the former is used for synchronization and the latter provides information about the rate used to transmit the remaining portion of the PPDU.

Additional overhead is introduced by the MAC and the LLC/SNAP headers. The Subnetwork Access Protocol (SNAP) is an extension to the IEEE 802.2 LLC and provides a mechanism for multiplexing different upper layer protocols. SNAP supports identifying protocols by EtherType[2] field values. SNAP headers are introduced since

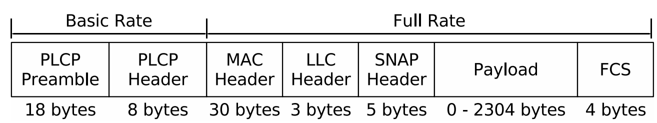


Fig. 9. IEEE 802.11 encapsulation.

the IEEE 802.11 MAC header does not have an EtherType field, to indicate which protocol is being transported.

In the IEEE 802.11 protocol, the mechanism used to access the wireless medium is called Distributed Coordination Function (DCF). According to the DCF scheme, a station that wants to transmit a packet monitors the channel until an idle period equal to the Distributed Inter-Frame Space (DIFS) is detected. Then, the station generates a random back-off counter. The back-off counter is decremented as long as the channel is idle, frozen when a transmission is detected, and reactivated when the channel is sensed free for a DIFS interval. The station transmits when the back-off counter time reaches zero. Initially, the back-off timer is drawn uniformly in the back-off window $[0, CW_{\min}]$: at each retransmission (due to collisions or errors), the back-off window is doubled until it reaches the maximal size $2^m CW_{\min}$. Thus, due to the back-off procedure, the concurrent transmission of alien stations affects the service time of IEEE802.11: the higher the number of transmitting stations, the larger the overhead [9]. We recall that half-duplex IEEE 802.11 stations cannot hear their own transmission. An acknowledgment (ACK) is then transmitted by the destination station to the source station if a packet transmission is successful. An ACK is always transmitted after a Short Inter-Frame Space (SIFS). If the transmitting station does not receive the ACK within an Extended Inter-Frame Space (EIFS) it re-schedules the transmission like if a collision occurred. The whole process is sketched in Fig. 10. The DCF scheme can be optionally enforced by a four-way handshaking technique, i.e. the

[2] The EtherType is a field used in the Ethernet frame to indicate which protocol is being transported, e.g. value 0×800 is associated with the IPv4 protocol.

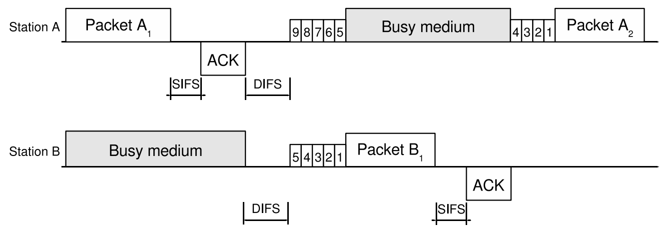


Fig. 10. IEEE 802.11 access scheme based on the DCF.

Request-to-Send/Clear-to-Send (RTS/CTS) handshake. The RTS/CTS scheme, being optional, is not implemented in most commercial cards, and in what follows we will then focus on the basic access scheme only.

6.2. Aggregation in error-prone channels

Aggregating multiple frames into a single burst at the MAC level allows to both reduce encapsulation and back-off overhead and increase the system throughput. The last claim is supported by the analytical model introduced in [7]. The author provides an accurate estimation of the IEEE 802.11 saturation throughput for both the basic and the RTS/CTS access scheme for an error-free channel. However, channel conditions in WMNs are not ideal and packets are lost due to collisions *and* transmission errors. As a result, even under light load conditions, routing metrics that minimize the hop count do not lead to good performances [26]: a two hop path over reliable and fast links can lead to better performances than a single hop route over a lossy and/or slow link.

Our adaptive aggregation scheme, in particular, exploits the Expected Transmission Count (ETX) metric as a cross-layer technique able to match routing and MAC layer parameters. We have chosen ETX because it is used as basic building block for other routing metrics (e.g. ETT [2,6], WCETT [6]) making our adaptive aggregation policy suitable for a wide rage of deployment scenarios. Here, we provided a brief introduction to ETX, for a detailed evaluation of the performance of different routing metrics, including ETX, we refer to [26].

ETX estimates the average number of retransmissions needed by each node to successfully deliver a packet over a given link. In order to compute the metric, each node periodically broadcast probes at data rates. Each probe contains the overall number of probes received from each of the neighbors during a specific observation window. Each station then computes the loss rate over a specific link (the IEEE 802.11 MAC does not retransmit the broadcast packets). Considering that a successful unicast transmission in IEEE 802.11 requires sending the data packet and receiving the corresponding ACK, the ETX metric between nodes A and B, can be computed as

$$M_{\text{ETX}} = \frac{1}{(1-P_{AB})(1-P_{BA})} = \frac{1}{P_{\text{Uni}}}, \tag{4}$$

where P_{AB} and P_{BA} are, respectively, the loss rate from A to B and from B to A and P_{Uni} the overall probability of a successful unicast transmission. Being P_e the Frame Error Rate (FER) and p the collision probability, it follows

$$P_{\text{Uni}} = (1-p)(1-P_e). \tag{5}$$

Assuming also that errors after decoding are i.i.d. over the frame bits, and being P_b the Bit Error Rate (BER), is stands:

$$P_e = 1-(1-P_b)^L. \tag{6}$$

By combining (4)–(6), we obtain

$$P_b = 1 - \mathrm{e}^{-\frac{\log[M_{\text{ETX}}(1-p)]}{L_{\text{Probe}}}}, \tag{7}$$

where L_{Probe} is the length of the probe packet used for computing M_{ETX}. The network throughput computed according to the Bianchi's model presented in [7] can then be extended to account for error-prone channels

$$\begin{aligned} S &= \frac{P_s(1-P_e)E[P]}{\sigma P_i + P_s P_e T_e + P_s(1-P_e)T_s + (1-P_c)T_c} \\ &= \frac{P_s(1-P_e)E[P]}{\sigma P_i + P_s T_s + P_c T_c} = \frac{AB^L L}{C+DL}, \end{aligned} \tag{8}$$

where for the ease of reading we put

$$A = \frac{P_s}{R}, \quad B = 1-P_b, \tag{9}$$

$$C = \sigma P_i + P_s T_{s_0} + P_c T_{c_0}, \quad D = \frac{P_s + P_c}{R}. \tag{10}$$

Also, we denoted

$$T_{s_0} = H + \text{SIFS} + \delta + \text{ACK} + \text{DIFS} + \delta \tag{11}$$

$$T_{c_0} = H + \text{DIFS} + \delta \tag{12}$$

$$T_s = T_{s_0} + E[P], \quad T_c = T_{c_0} + E[P^*], \tag{13}$$

being and T_e the time the channel is sensed busy for a failed transmission due to a channel error. Fig. 11 plots the system saturation throughput vs. an increasing length of the payload for different values of the routing metric ETX. As we can see, the argument that is the basis for the packet

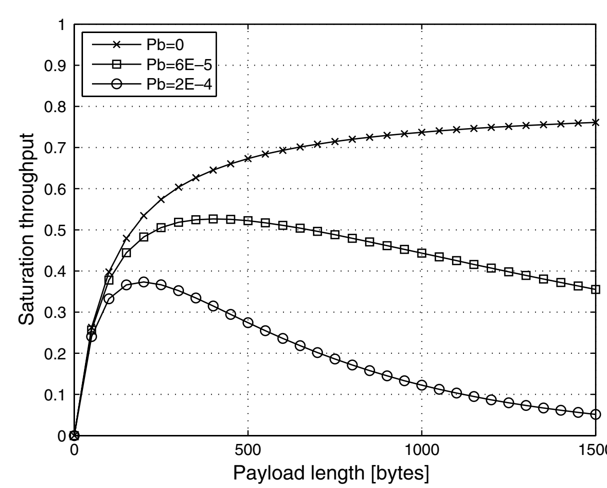


Fig. 11. IEEE 802.11 saturation throughput vs. an increasing length of the payload for different values of the ETX metric.

aggregation technique described in the following is that if we increase the payload length, we decrease system overhead. But, there exist a trade off, since frame errors become more likely with the packet length, i.e. the number of aggregated packets. Assuming as a first approximation constant length packets, the optimal packet length L_{Opt} writes as

$$f(M_{\text{ETX}}) = -\frac{C}{2D}\left(1 - \sqrt{1 + \frac{4DL_{\text{Probe}}}{C\log M_{\text{ETX}}(1-p)}}\right). \quad (14)$$

6.2.1. Run-time measurements and adaptation

Our aggregation scheme exploits the *monitor mode* provided by the Atheros-based WiFi interfaces building our testbed. Madwifi [27] is used as device driver. The monitor mode, or RFMON mode, allows a computer with a wireless NIC to monitor all traffic received from the wireless network, which gives us a good estimation of the channel load since it considers also traffic from other WiFi sources not participating the WMN. The monitor mode is similar to the promiscuous mode used for packet sniffing in wired networks. This feature has been exploited to monitor the number of active neighbors in the last N seconds. This parameter is used in order to compute both the p and τ values that appear in the Bianchi's model [7], and the C, and D parameters used in (14). In order to speed-up computation, a look-up table is used. Numerical values are pre-computed for $1 \leqslant N \leqslant 30$.

A typical drawback of a packet aggregation scheme is that it increases the processing delay at each node and becomes less suitable for delay sensitive applications (e.g. VoIP). Moreover, (14) was derived under saturation conditions, and the computed optimal burst size will affect the transmission delays if the offered traffic is low. In [14] an additional parameter representing the channel utilization is used in order to module the burst length according to the channel utilization. A node measures the channel *busy* when sending a frame, another node is transmitting or nodes network allocation vector indicates that the channel is reserved. However, using off-the-shelf IEEE 802.11 interfaces, the packet aggregation scheme implemented in our testbed cannot extract such information from the MAC layer. We have then decided to indirectly estimate the channel occupation exploiting the *monitor mode* provided by the Atheros-based WiFi interfaces building our testbed. The monitored parameters are:

- Exponential moving average of both packet rate (R) and per-packet airtime (A), A is computed as the product of packet length (bits) and transmission rate (Mbit/s) plus PLCP headers duration.
- Number of active neighbors in the last N seconds.

An estimate of the channel load (μ) is produced every N seconds, as the ratio between the air time and the time between two successive transmissions over the channel

$$\mu = \frac{A}{(1/R)} = AR. \quad (15)$$

Using (14), we then approximate the optimal burst length as follows:

$$L_{\text{Opt}} = \min[\mu f(M_{\text{ETX}}), B_{\text{Max}}], \quad (16)$$

where B_{Max} is the maximum frame size supported by the local area network technology. This parameter has been set to 1500 bytes which is the MTU (Maximum Transmission Unit) supported by an Ethernet LAN. Based on (16), and estimating the parameters described before, a node will then be able to update L_{Opt} according to the link conditions.

6.2.2. Hop-by-hop packet aggregation

According with our architecture, packets aggregation and de-aggregation is performed at each hop. Albeit such an approach could lead to increasing delays as the number of hops increases, we postulate that, at intermediate nodes, medium access delay is sufficient to collect enough packets so that burst generation is triggered by the optimal frame length without incurring in any aggregation delay.

Algorithm 1. Hop-by-hop burst generation policy:

```
1: if size(queue) ⩾ L_Opt then
2:    if size(queue) ⩽ B_Max then
3:       generate a burst no longer than L_Opt bytes
4:    else
5:       generate a burst no longer than B_Max bytes
6:    end if
7: else if Aggregation timer is expired then
8:    aggregate all the packets in the queue
9: end if
```

6.3. Evaluation

Our measurements campaign aimed at evaluating the number of concurrent VoIP flows that can be sustained, i.e. the voice capacity of the system. Three different setups are considered. The first setup (*Bulk*) does not use any aggregation policy and is used as benchmark. The second setup (*Aggregator*) exploits our adaptive aggregation policy, however, channel load information is not utilized ($\mu = 1$). The third setup (*AdjAggregator*) exploits the channel load information in order to modulate the optimal burst length according to (16). During this measurements campaign the aggregation timer has been set to 20 ms.

During our measurements campaign we looked at delay, packet loss, and R-factor as main QoS metrics. In the rest of this Section we will focus our attention on the 2-hops route terminated at node number 3. We expect the mean packet delay to be higher when our aggregation policy is applied without channel load correction in particular for a low number of concurrent VoIP calls. This is confirmed

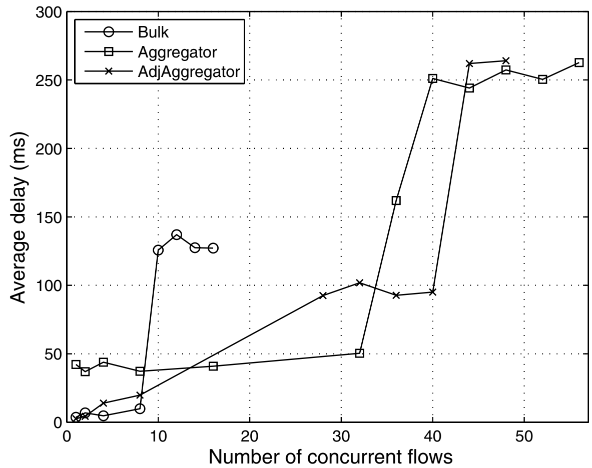


Fig. 12. Average delay vs. number of concurrent flows.

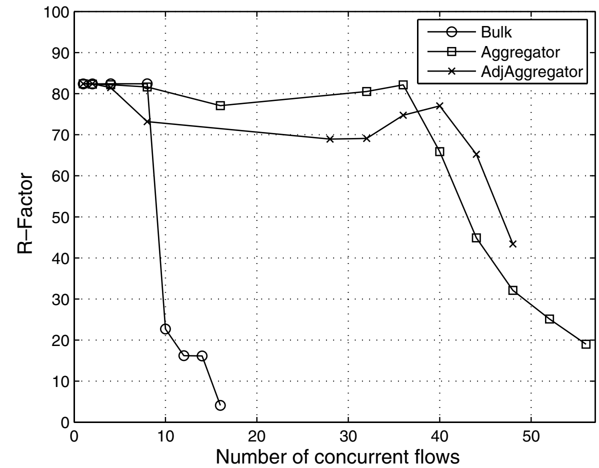


Fig. 14. Average *R*-factor vs. number of concurrent flows.

from the results plotted in Fig. 12, which reports on the average delay vs. number of concurrent flows. Packet loss and *R*-Factor are plotted, respectively, in Fig. 13 and in Fig. 14. Being n the number of concurrent VoIP flows, we can roughly identify three zones:

(1) $n \leqslant 8$, the three setups show the same performances;
(2) $8 < n < 32$, the *Bulk* setup has reached its maximum number of concurrent voice sessions, while the two aggregation schemes are still able to sustain additional VoIP calls with the *AdjAggregator* showing slightly worse performances in terms of both delay and packet loss;
(3) $n \geqslant 32$, both aggregation schemes have reached their maximum number of voice session.

Overall, both aggregation policies provides at least a factor 4 performance increase, since the number of sustained sessions reaches 32, whereas the plain IEEE 802.11 protocol allows for just 8 VoIP sessions. The *AdjAggregator* trades low-latency for small number of VoIP calls with a slightly higher packet loss when the number of concurrent VoIP sessions increases. We ascribe that behavior to a negative feedback involving our channel load driven packet aggregation policy, that finally leads to an oscillation in the optimal frame size and, as a consequence, to an alternation between burst of short packets and burst of long packets.

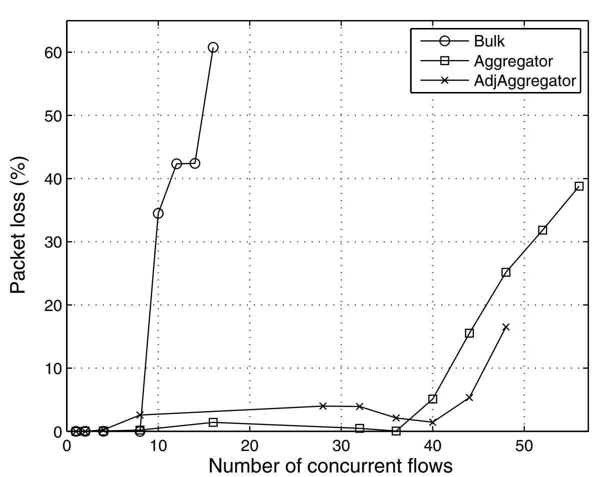


Fig. 13. Average packet loss vs. number of concurrent flows.

Finally, we have explored the impact of the aggregation timer on the QoS figures. This scenario exploits a star-shaped network topology (see Fig. 15) where all nodes are in radio range. Fig. 16 reports on the outcome of the measurements. As we can see, when the number of concurrent flows is lower than 13, the burst is triggered by the aggregation timer thus lowering its value leads to better performances. On the other hand, when the network load

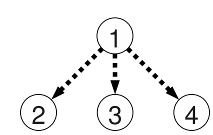


Fig. 15. Star-shaped network topology exploited in order to evaluate the impact of the aggregation timer.

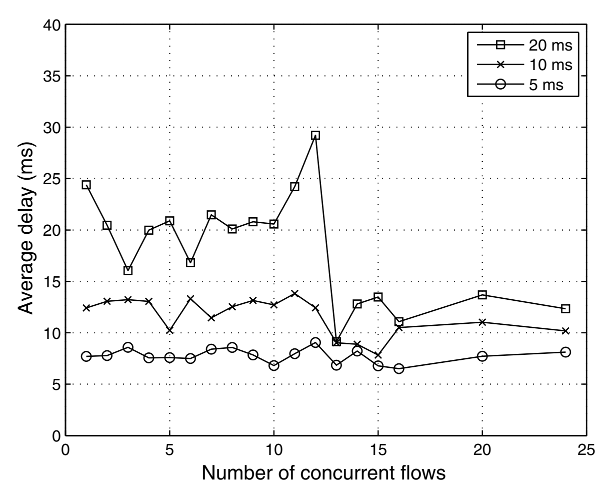


Fig. 16. Delay vs. number of concurrent flows for different values of aggregation timer.

increases (more than 13 concurrent flows) the performances do not depend on the timer's value in that the burst is triggered by the optimal burst length formula. It is worth noting that, for low values of the aggregation timer (5 ms,10 ms), the average delay remains constant. This behavior is due to the fact that when the aggregation delay is low compared to the packet inter-arrival time and the optimal burst size is never reached within the window given by the aggregation delay.

7. Conclusions

In this paper, we have proposed a lightweight architecture combining service differentiation with a novel packet aggregation policy in IEEE 802.11-based WMNs. The proposed scheme has been implemented as an extension of the MIT Roofnet platform and its performances have been tested over a WiFi-based testbed. Results show that the proposed technique is able to effectively differentiate services and improve the network scalability. With respect to the specific case of VoIP traffic, in fact, we experienced a factor 4 increase in the number of simultaneous VoIP sessions. The software implementation of the proposed mechanism has been released under an open-source license, with the aim of providing the reference scientific community with a basis for developing further innovative solutions for service differentiation in WMNs.

Among the various possible research directions to be pursued to extend the current work, two appear more promising. The first one is based on the use of IEEE 802.11e-compliant cards. In this case, indeed, the layer 2.5 priority classes could be mapped to the service classes offered by the underlying MAC protocol, achieving a better differentiation of services with respect to what happens in our framework. The second one involves investigating the adaptive aggregation policy self-interference generated by the channel load driven burst modulation.

References

[1] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Elsevier Computer Networks 47 (4) (2005) 445–487.

[2] J. Bicket, D. Aguayo, S. Biswas, R. Morris, Architecture and evaluation of an unplanned 802.11b mesh network, in: Proceedings of ACM MOBICOM, Cologne, Germany, 2005.

[3] J. Li, C. Blake, D.S.J.D. Couto, H.I. Lee, R. Morris, Capacity of ad hoc wireless networks, in: Proceedings of IEEE/ACM MobiCom, Rome, Italy, 2001.

[4] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks, in: Proceedings of BroadNets, San Jose, California, USA, 2004.

[5] D.S.J.D. Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proceedings of ACM MobiCom, San Diego, California, USA, 2003.

[6] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proceedings of ACM MOBICOM, Philadelphia, Pennsylvania, USA, 2004.

[7] G. Bianchi, IEEE 802.11 – saturation throughput analysis, IEEE Communication Letters 2 (12) (1998) 318–320.

[8] J. Yin, X. Wang, D. Agrawal, Optimal packet size in error-prone channel for IEEE 802.11 distributed coordination function, in: Proceedings of IEEE WCNC, Atlanta, Georgia, USA, 2004.

[9] F.D. Pellegrini, F. Maguolo, A. Zanella, M. Zorzi, A cross-layer solution for VoIP over ieee802.11, in: Proceedings of WPMC 2005, Alborg, Denmark, 2005.

[10] W. Wang, S.C. Liew, V.O.K. Li, Solutions to performance problems in VoIP over a 802.11 wireless LAN, IEEE Transaction on Vehicular Technology 54 (1) (2005) 366–384.

[11] D. Kliazovich, F. Granelli, Packet concatenation at the IP level for performance enhancement in wireless local area networks, in: ACM/Springer Wireless Networks.

[12] K. Lu, D. Wu, Y. Fang, A novel framework for medium access control in ultra-wideband ad hoc networks, dynamics of continuous, Discrete and Impulsive Systems 12 (3) (2005) 427–441.

[13] S. Ganguly, N.V.K. Kim, A. Kashyap, D. Niculescu, R.I.S. Hong, S. Das, Performance optimization for deploying VoIP services in mesh networks, IEEE Journal on Selected Areas in Communications 24 (11) (2006) 2147–2158.

[14] K. Lu, J. Wang, D. Wu, Y. Fang, Performance of a burst-frame-based CSMA/CA protocol for high data rate ultra-wideband networks: analysis and enhancement, in: Proceedings of ACM QShine, Waterloo, Ontario, Canada, 2006.

[15] M. Katsevenis, S. Sidiropoulos, C. Courcobetis, Weighted round-robin cell multiplexing in a general-purpose ATM switch chip, IEEE Journal on Selected Areas in Communications 9 (8) (1991) 1265–1279.

[16] IETF, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Available from: <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt>.

[17] E. Kohler, B.C. Robert Morris, J. Jannotti, M.F. Kaashoek, The click modular router, ACM Transaction on Computer System 18 (3) (2000) 263–297.

[18] Jtg. <https://hoslab.cs.helsinki.fi/savane/projects/jtg/>.

[19] ITU-T Recommendation G.729 Annex B, A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70, 1996.

[20] ITU-T Recommendation G.107, The E-model, a computational model for use in transmission planning, 2005.

[21] Simple Network Time Protocol (SNTP) Version 4. Available from: <http://www.apps.ietf.org/rfc/rfc2030.html>.

[22] IEEE-802.11WG, IEEE 802.11e Standard Draft/D8.0: Draft Supplement to Standard for Telecommunications and Information Exchange between Systems LAN/MAN Specific Requirements Part 11: MAC enhancements, 2004.

[23] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, RFC 2475: an architecture for differentiated services (1998). Available from: <http://tools.ietf.org/html/rfc2475.txt>.

[24] S. Lu, V. Bharghavan, R. Srikant, Fair scheduling in wireless packet networks, IEEE Transactions on Network 7 (4) (1999) 473–489.

[25] IEEE Standard for Local and Metropolitan Area Networks Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[26] R. Draves, J. Padhye, B. Zill, Comparison of routing metrics for static multi-hop wireless networks, in: Proceedings of ACM SIGCOMM, Portland, Oregon, USA, 2004.

[27] Madwifi. Available from: <http://madwifi.org/>.