Seeking for an Optimal Route in IEEE 802.11e Ad-Hoc Networks

João M. Ferro*, Orlando Cabral* and Fernando J. Velez*,†
*Instituto de Telecomunicações, Universidade da Beira Interior
Calçada Fonte do Lameiro
6201-001 Covilhã, Portugal
†Centre for Telecommunications Research, King's College London
Strand
London WC2R 2LS, UK
Email: {ferro, ombc}@lx.it.pt, fjv@ubi.pt

Abstract—**In this paper we present several different approaches to a routing calculation in an ad-hoc network. By using different weights for each link, each algorithm generates a different path, with its own number of hops and maximum throughput. Tests were performed in our own IEEE 802.11e simulator and the results allow to conclude that the cross-layer approach has advantages over the traditional one. In fact, our proposals delivers up to three times more packets than the standard one, whilst diminishing the end-to-end delay. Results show that the proposal which privileges users with a baud rate near the maximum possible achieves better performance than the others. A baud rate near the maximum corresponds to links with the highest signal to interference-plus-noise ratio. The proposed technique privileges the use of paths with a larger number of hops, and reduces the packet end-to-end delay.**

I. INTRODUCTION

In ad-hoc networks packet forwarding is a very important aspect. Stations that generate the packets must know to where they should forward their data, and this must be done efficiently. Since IEEE 802.11 only regulates the Medium Access Control (MAC) and Physical (PHY) layers, the routing principles are not covered by the standard. Several proposals have been made in this field. For example, [1] has demonstrated that a cross-layer approach can reach better results than a standard one. The authors from [2] present the Expected Transmission Count (ETX) metric, which attributes to every link a cost related to the packet delivery ratio. The Expected Transmission Time (ETT) was presented in [3] and is a function of the loss rate and the bandwidth of the link. In this work, we use new cross-layer approaches that inject values read from the Physical and MAC layers into the routing component, in a search for the best possible path. By introducing this information, we dynamically generate the network routing table, which is changed according to the load of the stations, trying to use the best possible path. If the best path is congested, then an alternative one may be chosen.

This paper is organised as follows. Section II presents a brief and basic explanation about the IEEE 802.11e standard. Section III describes the routing techniques based on cross-layer design, used in this paper. Section IV presents the results of our simulations. Finally, in Section V we make the considerations about the results, presenting the conclusions and the undergoing work.

II. IEEE 802.11E

The IEEE 802.11 is a family of protocols designed to standardise the transmission of packets in a Wireless Local Area Network (WLAN). It provides regulation for the MAC and PHY layers, only describing how the direct communication between two stations is performed. The IEEE 802.11e is an amendment to the original standard, and is built for Quality-of-Service support. Since the original IEEE 802.11 treats all the packets as equals, the most time-sensitive traffic – such as video and voice – could not arrive on time at destination. The improvement brought by this amendment is the classification of the packets into four different classes – video (VI), voice (VO), best effort (BE) and background (BK), and the assignment of different priorities to each class or access category (AC). This is accomplished at the MAC layer, where a new function is introduced. The original IEEE 802.11 uses a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) methodology, denominated Distribution Coordination Function (DCF), to coordinate the access of the stations to the shared medium. This amendment introduces a new function called Hybrid Coordination Function (HCF), which uses the Enhanced Distributed Channel Access (EDCA) and the HCF Controlled Channel Access (HCCA). The EDCA provides a contention-based access to the channel, while HCCA provides access to the channel in a contention-free manner. In both methods each station tries to gain a Transmission Opportunity (TXOP), which will allow it to transmit a packet.

The IEEE 802.11e defines two types of networks, one called Basic Service Set (BSS), in which there is an Access Point (AP) that acts like the coordinator of the network, all stations in the network are controlled by it and can communicate only with the AP. And another, called Independent Basic Service Set (IBSS), in which there is no AP and all the stations can communicate among them, as long as they are within the range of their radio device. In this work we consider only the latter, which can also be called an ad-hoc network. Besides, only EDCA is considered.

III. ROUTING TECHNIQUES

The experiences were performed by using the IEE 802.11e simulator that was developed at Instituto de Telecomunicaçõe. It is not an objective of this paper to provide a full description of the features of this simulator; for that one can refer to [4]. We used a modification to the original simulator, described in [5], to enable a multi-hop environment. The routing calculation is performed by running Dijkstra's algorithm in a table containing the cost of the links between all the stations in the network. The algorithm then generates a table in which for each pair source/destination there is the indication of the next hop. To determine the best approach, we use information gathered from the PHY layer to determine the cost of each link, and perform experiences with different techniques. The system is tested for the cost functions presented in Table I.

TABLE I
COST CALCULATION FORMULA FOR EACH ALGORITHM

Alg.	Link cost
No	1
A	$\sqrt{(20 \times 10^6 - link\_throughput)^2}$
B	$\sqrt{(21 \times 10^6 - link\_throughput)^2}$
C	$\sqrt{(30 \times 10^6 - link\_throughput)^2}$
D	$\sqrt{(56 \times 10^6 - link\_throughput)^2}$

The $link\_throughput$ is acquired from the MAC layer of the simulator, and varies according to the Signal to Interference-plus-Noise Ratio (SINR) measured at the sender, in accordance with data from [6]. The D approach may be considered as a greedy one, where the routing procedures choose the path that offers higher data rate, i.e, whose SINR is higher and in which stations are closer to each other. Note that a larger number of hops is considered with this metric (to arrive to a destination). As the cost algorithm goes from D to A, the routing protocol considers connections with lower SINR but also lower number of hops.

To achieve better performance, we update the routing table by looping back to the number of collisions that occur at a certain station. The reason to do this is very simple: as the algorithms will always look for the best path possible, if a certain station is included in the route of several best paths, it may become overloaded with packets. By becoming overloaded, it will not be able to process and deliver all the packets it receives, thus it will start dropping packets, reducing the delivery rate. Another side effect is the following: since its buffer is fully occupied with packets, a packet will take longer time to be transmitted (since it is waiting in the queue for its opportunity), which increases the end-to-end delay. By introducing this metric to each link, the number of collisions that occurred at the receiver station are added (meaning that on the links $1 \rightarrow 2$ and $2 \rightarrow 1$ the algorithm will pick up the collisions that occurred at stations 2 and 1, respectively – link asymmetry is a possibility). A station that is receiving more packets than it can deliver will have an higher number of collisions. The cost of the paths that use it will possibly increase, until it reaches a point in which another path will have a lower cost and packets will be diverted to that one. To test this new approach we have kept the same philosophy as before, i.e., we have kept algorithms A, B, C and D and added the collisions multiplied by a weight. The respective formulas are presented in Table II.

TABLE II
COST CALCULATION FORMULAS WHEN COLLISIONS ARE CONSIDERED

Alg.	Link cost
A'	$\sqrt{(20 \times 10^6 - link\_throughput)^2} + \left(1 \times 10^5 \times collisions\right)$
B'	$\sqrt{(21 \times 10^6 - link\_throughput)^2} + \left(1 \times 10^5 \times collisions\right)$
C'	$\sqrt{(30 \times 10^6 - link\_throughput)^2} + \left(1 \times 10^5 \times collisions\right)$
D'	$\sqrt{(56 \times 10^6 - link\_throughput)^2} + \left(1 \times 10^5 \times collisions\right)$

The number of collisions that a station is experiencing may not be the correct way of evaluating if a station is working beyond its capabilities or not. For example, if station A experiences 10 collisions and station B just 1, one could think that station A is working more overloaded than station B. However, if it is said that algorithm A managed to deliver 100 packets and B just 8, we see clearly that station A is not overloaded. For this reason, we introduce a metric called collision rate (we will denote it in the formulas as $collision\_rate$), which is computed as the ratio between the number of collisions and number of packets successfully transmitted by a machine. Our approach will be similar to the one presented previously but each link will now include this new function, which is computed according to the values on the receiver machine (number of packets sent successfully, and collisions occurred). The respective formulas are presented in Table III.

TABLE III
COST CALCULATION FORMULAS WHEN THE COLLISION RATE IS CONSIDERED

Alg.	Link cost
A"	$\sqrt{(20 \times 10^6 - link\_throughput)^2} \times (1 + collision\_rate)$
B"	$\sqrt{(21 \times 10^6 - link\_throughput)^2} \times (1 + collision\_rate)$
C"	$\sqrt{(30 \times 10^6 - link\_throughput)^2} \times (1 + collision\_rate)$
D"	$\sqrt{(56 \times 10^6 - link\_throughput)^2} \times (1 + collision\_rate)$

The routing table is updated every time period P_1, taking into account the average status of the system in a period P_2, being $P_1 \geq P_2$. Several functions like exponentials, and potentials with different weights were considered. If more weight is given to the parameter collision rate, the traffic ends to be redirected through the same hop, which ends up by causing an overloaded hop (having more traffic than the one that the hop can manage) suffering more collision and packet losses. Having low weight causes no difference in managing the load among hops. The function proposed is not the optimal solution to account for the collisions, but from the ones tested, it is the one that presents better results.

IV. RESULTS

To test the performance of these routing algorithms, we deployed randomly 30 stations on a 2D spatial field with uniform distribution (by using a pseudo-random number generator), and started generating traffic among them. Traffic is generated as described in Table IV.

TABLE IV
THE TRAFFIC GENERATED FOR THE TESTS

Traffic type	Source	Destination
VI	3	10
VO	11	12
VO	12	13
VO	13	14
BK	3	4
BK	4	5
BK	5	6

The parameters used to configure our simulator are presented in Table V, in which the Short Interframe Space is abbreviated to SIFS, the Arbitration Interframe Space to AIFS, and acknowledgement to ACK.

TABLE V
SIMULATION PARAMETERS

Parameter	Value
Inter arrival time for VI	10ms
Inter arrival time for VO	20ms
Inter arrival time for BK	12.5ms
VI payload	10240b
VO payload	1280b
BK payload	18430b
Simulation time	15s
Field of test	$120\times120\text{m}^2$
Stations deployed	30
SIFS	16μs
AIFS (VI)	SIFS+2×Slot time
AIFS (VO)	SIFS+2×Slot time
AIFS (BK)	SIFS+7×Slot time
Slot time	9μs
ACK payload	112b

We have chosen a simulation time of 15s, while the dynamic routing table is computed in intervals of 1s and the simulation runs in the same conditions 10 times for each algorithm (the results presented in this paper are an average of these results). The study area is a square field with 120m of side, where the 30 stations are placed according to the deployment from Fig. 1.

For each algorithm we kept track of the paths selected for each flow. We present these results for the video stream in Table VI.

From the analysis of the routing table, the first conclusion is that the use of the information on the number of collisions causes a lot of instability in the routing, forcing the routing table to change paths very frequently. For example, in the algorithm A' the path is changed 7 times (from the original one), and by looking at the time log (it is not presented in this paper because it is very long) we notice that it changes every second. Algorithms that use the metric collision rate change the routing table less times; in fact B" and D" do not change it at all, while A" is only changed once (we may add that at the first second to path is changed and remains the new one until the end of the simulation). Nevertheless, since stability does not mean the best performance, we present the results for the most important metric. In our case, we consider that the most important aspect is that packets reach destination, and reach it in a reasonable time. As so, we will consider the number of packets delivered as the indicator for the best performance (more packets = better algorithm), and weight this with the end-to-end delay (an algorithm that delivers more packets but has a longer delay is not considered very good). The results for the metrics obtained for each algorithm are presented in Fig. 2 and in Table VII.

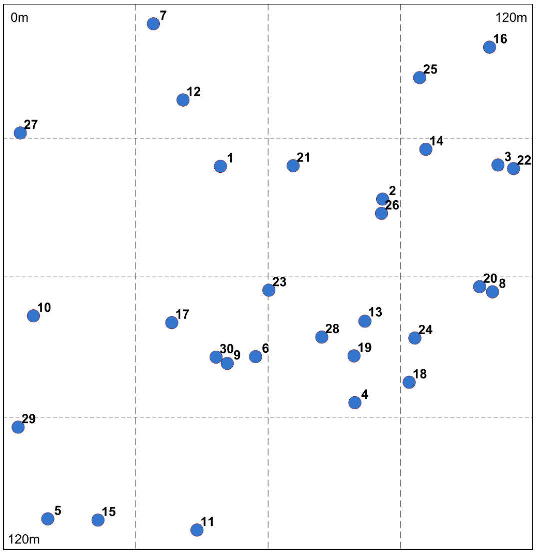

Fig. 1. The deployment of the stations for the test

By analysing these results one can first conclude that the cross-layer approach always achieves better results than the normal one, both in terms of the end-to-end delay (it is lower in all cases) and in the number of packets delivered at destination (it delivers more packets in any case). By comparing

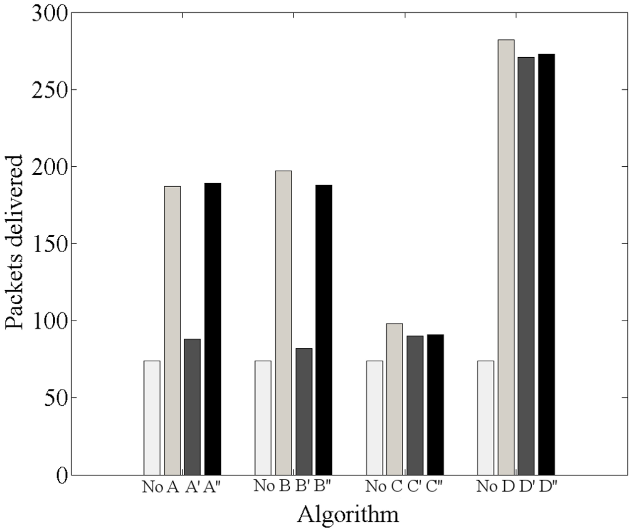

Fig. 2. Video packets delivered for the metrics proposed.

TABLE VI
THE ROUTING TABLE CHANGES DURING OUR SIMULATION

Algorithm	Paths used to connect station 3 to 10
No	3 26 28 17 10
A	3 26 13 6 17 10
B	3 26 13 23 17 10
C	3 26 23 17 10
D	3 20 24 13 28 6 30 17 10
A’	3 26 13 6 17 10
	3 20 19 6 17 10
	3 2 23 17 10
	3 8 18 28 17 10
	3 20 18 28 17 10
	3 2 28 17 10
	3 16 14 21 23 17 10
	3 2 28 17 10
B’	3 26 13 23 17 10
	3 2 23 17 10
	3 8 18 28 17 10
	3 20 18 28 17 10
	3 16 14 21 23 17 10
C’	3 26 23 17 10
	3 2 23 17 10
	3 8 24 28 6 17 10
	3 2 28 6 17 10
	3 2 28 9 17 10
	3 20 18 28 6 17 10
	3 8 18 28 9 17 10
	3 20 24 28 9 17 10
D’	3 20 24 13 28 6 30 17 10
	3 2 23 6 30 17 10
	3 26 23 17 10
	3 2 28 6 30 17 10
A”	3 26 13 6 17 10
	3 26 13 23 17 10
B”	3 26 13 23 17 10
C”	3 26 23 17 10
	3 2 28 6 17 10
	3 8 24 28 6 17 10
	3 2 23 17 10
	3 14 26 23 17 10
D”	3 20 24 13 28 6 30 17 10

our proposed algorithms based only in the raw throughput of the link, the one that privileges the use of links with higher bit rate can deliver more packets, with lower delay, despite of using paths with larger number of hops. This behaviour can also be observed in the other two cases: algorithms D’ and D” have a better performance than A’, B’, C’, and A”, B”, C”, respectively. By analysing the results that are achieved by introducing the collision metric in the routing calculation, one can conclude that this was not a good choice. Like stated above, this introduces a lot of instability in the routing table, which, in turn, degrades the network performance by reducing the number of delivered packets when directly compared with the previous approach (meaning, comparing A’ with A). With the inclusion of the new metric, i.e., the collision rate, the obtained results are better than the ones with just the collision, but slightly inferior than the other ones. This shows that an approach that considers the collision rate is promising because it can select new paths in case some of them get congested. This approach requires however further optimisation work, to achieve even a better performance.

TABLE VII
THE RESULTS FOR THE VIDEO STREAM

Algorithm	Delay [ms]	Packets delivered
No	6547	74
A	5416	187
B	5675	197
C	5704	98
D	5361	282
A’	4333	88
B’	4631	82
C’	6304	90
D’	5653	271
A”	5725	189
B”	5674	188
C”	5367	91
D”	5521	273

V. CONCLUSION

In this work some tests were performed in our custom-made IEEE 802.11e simulator by using 12 new different routing techniques. The objective was to understand which one would present an overall better performance. From the results on our wireless nodes deployment, all the new approaches presented in this work achieve better performance than a standard routing algorithm that minimizes the number of hops. From theses approaches, the one that privileges stations that are closer to each other achieves better performance than the others. Despite of using a larger number of hops, the proximity of the stations allows for the transmission to take place shortly (higher bit rate) and strongly (higher SINR). As the inclusion of other metrics to avoid paths with a high number of collisions is not having the desired effects, we are currently working on improving this aspect in order to achieve better results. As a future work, we intend to optimise our algorithm that uses the collision rate metric by changing the weight of this metric in the path calculation formula. For this purpose, we will apply genetic algorithms to seek for an optimal formula. We also intend to generalise our study by generating networks with random configurations while evaluating the general performance of our algorithms.

REFERENCES

[1] Y. Lu, H. Zhu, and Q. Wu, “A multi-paths routing solution based on srmpolsr and mac layer load,” in *IWCLD ’07, International Workshop on Cross Layer Design 2007*, Sept. 2007, pp. 117–120.

[2] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom ’03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003, pp. 134–146. [Online]. Available: http://dx.doi.org/10.1145/938985.939000

[3] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *MobiCom ’04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 114–128.

[4] O. Cabral, A. Segarra, and F. Velez, “Event-driven simulation for ieee 802.11e optimization,” *IAENG International Journal of Computer Science*, vol. 35, no. 1, pp. 161–173, 2008.

[5] J. Ferro and F. Velez, “Cross-layer multi-hop simulator for ad-hoc ieee 802.11e,” in *Proceedings of the 11th International Symposium of Wireless Personal Multimedia Communications (WPMC) 2008*, Lapland, Finland, Sep 2008.

[6] A. Grilo, M. Macedo, and M. Nunes, “A scheduling algorithm for qos support in ieee802.11 networks,” *Wireless Communications, IEEE*, vol. 10, no. 3, pp. 36–43, 2003. [Online]. Available: http://dx.doi.org/10.1109/MWC.2003.1209594