Improving TCP Performance in MANET by Exploiting MAC Layer Algorithms

Sofiane HAMRIOUI 1,2, Mustapha LALAM 2
1 USTHB, Algiers, Algeria
2 LARI, UMMTO, Tizi Ouzou, Algeria

Diyar K. Arab 3, Amine Berqia 3, Pascal LORENZ 4
3 ILAB, FCT-DEEI, Universidade Algarve Portugal
4 GRTC, UHA, IUT of Colmar, France

Abstract—In this paper, we propose an improvement of Transmission Control Protocol (TCP) protocol performance in MANET by exploiting the backoff algorithm of Medium Access Control (MAC) protocol. This improvement is IB-MAC (Improvement of Backoff algorithm of MAC protocol), it propose a new backoff algorithm based on a dynamic adaptation of its maximal limit according to the number of nodes and their mobility. The evaluation of our IB-MAC solution and the study of its incidences on TCP performance are done with AODV as like routing protocol, TCP New Reno as like transport protocol and varied network conditions such as load and mobility. The results are satisfactory and show that our algorithm can outperform not only MAC standard, but also similar techniques that have been proposed in the literature like MAC-LDA and MAC-WCCP.

MANET; TCP; MAC Layer; Algorithms; Performance.

I. INTRODUCTION

Mobile Ad Hoc Network (MANET) [1] is complex distributed systems that consist of wireless mobile or static nodes. In such networks, the MAC protocol [2], [3], [4] must provide access to the wireless medium efficiently and reduce interference. Important examples include Access Carrier-Sense Multiple Access (CSMA) with collision avoidance that uses a random back-off even after the carrier is sensed idle [5]; and a virtual carrier sensing mechanism using Request-To-Send/Clear-To-Send (RTS/CTS) control packets [6]. Both techniques are used in IEEE 802.11 MAC protocol [5] which is a current standard for wireless networks.

Transmission Control Protocol (TCP) [7], [8] is the transport protocol used in the most IP networks [9] and recently in ad hoc networks like MANET [10]. It is important to understand the TCP behavior when coupled with IEEE 802.11 MAC protocol [5] which is a current standard for wireless networks.

When the interactions between the MAC and TCP protocols are not taken into account, this may degrade MANET performance notably TCP performance parameters (like throughput and end-to-end delay) [11], [12], [13]. To adapt the behaviour of these two protocols to ensure better TCP performance and then better QoS [14], it is very important to study the interactions between them. In [15], we presented a study of interactions between the MAC and TCP protocols, we have shown that the TCP parameters performance (notably throughput and end-to-end delay) degrades while the nodes number increase in a MANET using IEEE 802.11 MAC as access control protocol. In [16], we have proposed solutions to the problem posed in [15], but we have just limited to a chain topology and also to the influence of the nodes number on the TCP performance.

Our contribution in this paper is the following of those done in [15], [16]. Other topologies have been studied and another parameter which is the nodes mobility has been considered. Also in this present work we compared our solution with other solutions proposed in the same context. After a short presentation of MAC and TCP protocols, we will present our IB-MAC (Improvement of the Backoff algorithm of MAC protocol) and study its incidences on TCP performance parameters (throughput and end-to-end delay). IB-MAC proposes a dynamic adaptation of the maximal limit of the MAC backoff algorithm. This adaptation is as function of the nodes number in the network and their mobility.

Our paper is structured in five sections: the section two gives a short presentation of MAC and TCP protocols, in section three we present the IB-MAC improvement to better TCP performance in MANET, in section four we study the incidences of these improvements on TCP performance parameters and we finish with section five which consists on conclusion and perspectives.

II. INTERACTIONS BETWEEN MAC ET TCP PROTOCOLS

A. MAC 802.11 and TCP Protocols in MANET

IEEE 802.11 MAC protocol defines two different access methods: polling based Point Coordination Function (PCF) and Distributed Coordination Function (DCF) which is used essentially in MANET.

The DCF access is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. When a node wants to transmit a frame, it senses the medium. If the medium is busy, it defers this transmission. If the medium is free for a specified time, called the Distributed
Inter-Frame Space (DIFS), the node is permitted to transmit. In order to avoid collision due to the hidden terminal problem [17], [18] the node first transmits a RTS control frame. The destination node responds with a CTS control frame. Both RTS and CTS frames include the duration of the transmission that will follow the RTS-CTS exchange. All nearby nodes receiving either the RTS or the CTS frame defer their pending transmissions for this duration. This deferral is referred to as virtual carrier sensing as it “senses” the medium through exchange of frames at the MAC layer. Once a successful RTS-CTS frame exchange takes place, the data frame is transmitted. The receiving node checks the received data frame, and upon correct receipt, sends an acknowledgement frame. If the sending node fails to receive the acknowledgement frame, it assumes that the data frame was lost. It backs-off and attempts a retransmission. After repeated failures to get the frame across, it simply drops the frame. Thus, although the introduction of RTS-CTS-DATA-ACK frame format makes the transmission more reliable, there is still the possibility of transmission failure. Such failures are more frequent in ad hoc networks than in MANET.

It has been shown that TCP does not work well in a wireless network [7], [19]. The wireless channel is subject to noise, so the losses packets are common. TCP associates the packet loss to the congestion, and then it starts its congestion control mechanism. Therefore, transmission failures at the MAC layer lead to the congestion control activation by TCP protocol then the number of packets is reduced (throughput). Several mechanisms have been proposed to address this problem [20], [21], [22], but most of them focus on the cellular architecture. The problem is more complex in multi-hop networks such as MANET where there is no base station and each node can act as a router [23], [24].

The TCP Performance parameters (like throughput and end-to-end delay) have been the subject of several evaluations. It has been shown that these parameters degrade when the interactions between MAC and TCP are not taken into account [7], [17]. In our previous work [15], we confirmed these results by studying the effect of the MAC layer when the number of nodes increases. The major source of these effects is the problem of hidden and exposed nodes [17], [18]. The most important solution which has been proposed to the hidden node problem is the use of RTS and CTS frames [25], [26]. Although the use of RTS/CTS frames is considered as a solution to the hidden node problem, it was shown in [15] [17] [27] that it also leads to further degradation of the TCP flow by creating more collisions and introduce an additional overhead which decrease the TCP performance.

B. Related Work

In [28] [29] [30] [31] [32], many analyses of TCP protocol performance are done and several solutions on how to improve its performance are proposed. In this subsection we present the most important solutions. Yuki et al. [33] have proposed a technique that combines data and ACK packets, and have shown through simulation that this technique can make radio channel utilization more efficient. The technique improved the TCP performance by up to 60% and by about 10% even when the network load was very high.

Altman and Jimenez [34], proposed an improvement for TCP performance by delaying 3-4 ACK packets. In their approach the receiver always delays 4 packets (except at the startup) or less if its timeout interval expires. The receiver uses a fixed interval of 100ms and does not react to packets that are out-of-order or filling in a gap in the receiver buffer, as opposed to the recommendation of RFC 1122.

Kherani and Shorey [35], suggest significant improvement in TCP performance as the delayed acknowledgement parameter $d$ increases to the TCP window size $W$. The novelty in approach is the analytic modeling of TCP over IEEE 802.11 based networks with a delayed acknowledgement parameter, and, a TCP window size of $W$ ($> 2$) packets.

Allman [36], conducted an extensive evaluation on Delayed Acknowledgment (DA) strategies, and they presented a variety of mechanisms to improve TCP performance in presence of side-effect of delayed ACKs.

Chandran [37] proposed TCP-feedback, with this solution, when an intermediate node detects the disruption of a route; it explicitly sends a Route Failure Notification (RFN) to the TCP sender. The source on receiving the RFN, it suspends all packet transmissions and freezes its state. But when a middle node learns of a new route to the destination, it sends a Route Re-establishment Notification (RRN) to the source.

Holland and Vaidya [38] proposed a similar approach based on ELFN (Explicit Link Failure Notification), when the TCP sender is informed of a link failure, it freezes its state. However, the source continues to send out packets at regular intervals to determine if a new route is available.

Liu and Singh [39] proposed the ATCP protocol; it tries to deal with the problem of high Bit Error Rate (BER) and route failures. The ATCP layer is inserted between the TCP and IP layers. ATCP puts TCP agent into the appropriate state after listening to the network state information provided by ECN (Explicit Congestion Notification) messages and by ICMP "Destination Unreachable" message. On receiving a "Destination Unreachable" message, TCP agent enters a persist state.

Fu et al. [40] investigated TCP improvements by using multiple end-to-end metrics instead of a single metric. They claim that a single metric may not provide accurate results in all conditions. They used four metrics: inter-packet delay difference at the receiver, short-term throughput, packet out-of-order delivery ratio, and packet loss ratio. These four metrics are cross checked for accurate detection of the network internal state.

Biaz and Vaidya [41] evaluated three schemes for predicting the reason for packet losses inside wireless networks. They applied simple statistics on observed Round-trip Time (RTT) and/or observed throughput of a TCP connection for deciding whether to increase or decrease the TCP congestion window. The general results were discouraging in that none of the evaluated schemes performed really well.

ISSN: 2249-9563
Liu et al. [42] proposed an end-to-end technique for distinguishing between packet loss due to congestion from packet loss by a wireless medium. They designed a Hidden Markov Model (HMM) algorithm to perform the mentioned discrimination taking RTT measurements over the end-to-end channel.

Kim et al. [43] proposed the TCP-BuS (TCP Buffering capability and Sequence information), like previous proposals, uses the network feedback in order to detect route failure events and to take convenient reaction to this event. The novel scheme in this proposal is the introduction of buffering capability in mobile nodes. The authors select the source-initiated on-demand ABR [44] (Associativity-Based Routing) routing protocol.

Oliveira and Braun [45] propose a dynamic adaptive strategy for minimizing the number of ACK packets in transit and mitigating spurious retransmissions. Using this strategy, the receiver adjusts itself to the wireless channel condition by delaying more ACK packets when the channel is in good condition and less otherwise.

Hamadani and Rakocevic [46] address the problem of TCP intra-flow instability in multihop ad hoc networks. They propose a cross layer algorithm called TCP Contention Control that it adjusts the amount of outstanding data in the network based on the level of contention experienced by packets as well as the throughput achieved by connections.

Zhai et al. [47] show that TCP suffers severe performance degradation and unfairness. Realizing that the main reason is the poor interaction between traditional TCP and the MAC layer, they propose a systematic solution named Wireless Congestion Control Protocol (WCCP) to address this problem in both layers. WCCP uses channel busyness ratio to allocate the shared resource and accordingly adjusts the sender’s rate so that the channel capacity can be fully utilized and fairness is improved.

Lohier and al. [48] proposes to adapt one of the MAC parameters, the Retry Limit (RL), to reduce the drop in performance due to the inappropriate triggering of TCP congestion control mechanisms. Starting from this, a MAC-layer LDA (Loss Differentiation Algorithm) is proposed. This LDA scheme is based on the adaptation of the RL parameter depending on the quality of the 802.11 wireless channel.

All the approaches presented above suggest improvements to TCP performance based either on MAC protocol or TCP protocol or on both. Our approach improve too the TCP performance, it is based on the backoff algorithm of MAC protocol. In what follows, we examine the interactions between MAC and TCP protocols before proceeding to the presentation of our solution.

III. IB-MAC (IMPROVEMENT OF THE BACKOFF ALGORITHM)

The MAC protocol is based on the Backoff algorithm that allows it to determine which will access to the wireless medium in order to avoid collisions. In the case of station finding a channel busy, the transmission is differentiated in accordance to the Backoff procedure whose principle is: as long as the channel is free for a DCF Inter Frame Space (DIFS) time (after a successful reception) or for an Extended Inter-Frame Space (EIFS) time (after a failed reception), the Backoff time is decreased. This time is calculated as follows:

\[ \text{BackoffTime} = \text{BackoffCounter} \times a\text{SlotTime} \]  (1)

In (1), a SlotTime is a time constant and BackoffCounter is an integer from uniform distribution in the interval [0, CW] and CW is the contention window who’s minimum and maximum limits are \((CW_{\text{min}}, CW_{\text{max}})\) and are defined in advance.

The CW value is increased in the case of non availability of the channel using the following formulas:

\[ m \leftarrow m + 1 \]
\[ CW(m) = (CW_{\text{min}} + 1) \times 2^m - 1 \]
\[ CW_{\text{min}} \leq CW(m) \leq CW_{\text{max}} \]

\( m \): the number of retransmissions.

The first parameter used by our IB-MAC solution is the number of nodes in the network. As we have seen through the simulations presented in the previous work [15] [16], when the number of nodes in the network increases, the performance of TCP deteriorates. The cause of this degradation is the frequent occurrence of collisions between nodes. More the number of nodes increases, more collisions are frequent. These collisions become more frequent with a small backoff interval because the probability to have two or more nodes choose the same value in a small interval is greater than the probability that these nodes choose the same value in a larger interval.

Note by \( I \) this interval, \( S_I \) its size, and \( Pr(i,x) \) the probability that the node \( i \) chooses the \( x \) value in the \( I \) interval.

The problem then is how to ensure that for any two nodes \( i \) and \( j \) in the network with \( i \neq j \), we will have:

\[ | Pr(i,x) - Pr(j,x) | = y \]
\[ \text{with } y \neq 0 \]  (3)

For an important number of nodes in the network, and for a high probability that the formula (3) will be verified, we must have a larger \( S_I \). To do this we have to make the size of \( S_I \) adaptable to the number of nodes in the network, then we intervene on one of the limits of this interval, we then propose the maximum limit \( CW_{\text{max}} \).

If \( n \) is the number of nodes in the network. Then the first part of the expression of \( CW_{\text{max}} \) will be:

\[ F(n) = \log(n) \]  (4)
Log (\(j\)) is used here because we found in [15] and [16] that the effects of the large values of the nodes number on the TCP performance are almost the same.

Our IB-MAC also takes into account the mobility of nodes. In fact, node mobility often leads to the breakdown of connectivity between nodes, resulting in loss of TCP packets and then the degradation of the TCP performance parameters (throughput and end-end delay). At the MAC protocol, when the packets losses are detected, they are associated to the collisions problem, which is not the case here. Therefore, we will try to find a compromise between the effect of mobility and the size of the backoff interval.

Mobility is generally characterized by its speed and angle of movement. These two factors determine the degree of the impact of mobility on packets loss. Consider a node \(i\), in communication with another node \(j\), then we note by:

- \(\alpha_i\): the angle between the line \((i, j)\) and the movement direction of node \(i\).

- \(W_i\): the speed of mobile node \(i\).

To consider the impact of mobility on the loss of packets is equivalent to considering the impact of its two parameters, \(W_i\) and \(\alpha_i\). For the effect of speed \(W_i\), as in the case of number of nodes, we use a logarithmic function because for large values of speed mobility the results converge. But when the node is static \((W_i=0)\) the effect of the mobility becomes zero, and then we must add 1 to the equation, so we will get:

\[
H'(W) = \log(W+1)
\]  
(5)

But in order to make the impact of this speed \(W\) positive on our backoff algorithm, we must change (5) as like follows:

\[
H(W) = 1/ (\log(W+1))
\]  
(6)

Also, the direction of node movement determines the degree of the influence of mobility on packets loss; it is given by \(G(W, \alpha)\) like follow:

\[
G(W_i, \alpha_i) = \begin{cases} 
1 & \text{if } \Pi/4 <= \alpha_i <= \Pi/4 \\
1/\sqrt{(W_i+1)} & \text{else}
\end{cases}
\]  
(7)

Note that \(G(W, \alpha) = 1\) when \(W = 0\) (without mobility). We added 1 to ensure that \(G\) will be defined for all the \(W\) values, and we used a ratio to get a positive effect of \(G\) on the backoff algorithm.

With (6) and (7) we can guarantee that when the mobility of nodes is significant, the adaptation of the backoff algorithm is not important because this mobility is more probable to be the cause of many losses packets. But with weak mobility the same equation makes it possible to get a significant adaptation to the backoff algorithm because in this case the collisions between frames are more probable to be the cause of the losses packets.

From (4), (6) and (7), we will have now the new expression of \(CW_{\text{max}}\) for node \(i\) as follows:

\[
CW_{\text{max}}(n, W_i, \alpha_i) = CW_{\text{max}0} + F(n) \times H(W_i) \times G(W_i, \alpha_i)
\]  
(8)

\(CW_{\text{max}0}\): initial \(CW_{\text{max}}\) defined by the MAC protocol (with the 802.11 version, it is equal to 1024).

Our approach is fully distributed within the MANET; each node may determine alone the values of \(n, W\) and \(\alpha\) so it can then calculate the value of \(CW_{\text{max}}\) according to the formula given in (8). The value of \(n\) is updated always when there is a new arrived node to the network or a leaved node from the network. So, our solution also contains an agent let updating the value of \(n\) as follows:

Variable \(N :=0;\)

......

Node\(_i\) := NEW (Node\(_Class\));

Add (Node\(_i\))

\(N:=N+1\)

......

Free (Node\(_j\))

\(N:=N-1;\)

After having made the values of CWmax adaptive to the number of nodes used and their mobility, the IB-MAC (improved version of that given by the formula (2)) becomes:

\[
m \leftarrow m + 1
\]

\[
CW(m) = (CW_{\text{min}}(n) + 1) \times 2^{m} - 1
\]  
(9)

\(CW_{\text{min}} <= CW(m) <= CW_{\text{max}}(n, W_i, \alpha_i)\)

With:

- \(m\): the number of retransmissions.

- \(n\): the number of the nodes used.

- \(\alpha_i\): the angle between the line formed by the mobile node and its corresponding node and the movement direction of this mobile node.

- \(W_i\): the speed of mobile node \(i\).

- \(CW_{\text{max}}\): initial value of \(CW_{\text{max}}\).

IV. EVALUATION OF IB-MAC AND ITS INCIDENCES ON TCP PERFORMANCE

A. Simulations Environment

The evaluation is performed through the simulation environment NS-2 (version 2.34) [49] from Lawrence Berkeley National Laboratory (LBNL) with wireless extension of CMU [50]. The MAC level uses the model 802.11b with...
the DCF (Distributed Coordination Function) which the values of its basic parameters are listed in the in TABLE 1. below.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble length (bit)</td>
<td>144</td>
</tr>
<tr>
<td>RTS length (bit)</td>
<td>160</td>
</tr>
<tr>
<td>CTS/ACK length (bit)</td>
<td>112</td>
</tr>
<tr>
<td>MAC header (bit)</td>
<td>224</td>
</tr>
<tr>
<td>IP header (bit)</td>
<td>160</td>
</tr>
<tr>
<td>SIFS (μ s)</td>
<td>10</td>
</tr>
<tr>
<td>DIFS (μ s)</td>
<td>50</td>
</tr>
<tr>
<td>Slot time (μ s)</td>
<td>20</td>
</tr>
<tr>
<td>Contention window</td>
<td>31</td>
</tr>
<tr>
<td>Retry limit</td>
<td>7</td>
</tr>
</tbody>
</table>

All nodes communicate through wireless links in half-duplex with an identical bandwidth of 1 Mb/s. For our simulations, the effective transmission range is of 250 meters and an interference range of 550 meters. Each node has a queue buffer link layer of 50 packets managed with a mode drop-tail [51]. The scheduling packet transmissions technique is the First in First out (FIFO) type. The propagation model used is the two-ray ground model [52].

Our simulations are done with reactive routing protocol AODV [53]. We used TCP NewReno [54] which is a reactive variant, derived and widely deployed, and whose performances were evaluated under conditions similar to those conducted here. This choice is because the previous work [55] has shown almost similar results for different routing protocols and TCP versions used. TCP traffic was used as the main traffic network. The different TCP variants are analyzed on the same topologies and the same pair source/destination are chosen by trial to ensure fairness and relevance of results.

The values, such as the duration of simulation, the speed of the nodes, and the number of connections have been established in order to obtain interpretable results compared to those published in the literature. The simulations are performed for 1000 seconds, this choice in order to analyze the full spectrum of TCP throughput.

We considered two cases without and with mobility. In the first case, chain topology is studied in which always the node 1 send for the node n (where n is the length of the chain). We just limited to the chain topology because our in our previous work [55] we have studied different topologies and we have found that it influence the communication environment in the same way. The distance between two neighboring nodes is 200 meters and each node can communicate only with its nearest neighbour. The interference range of a node is about two times higher than its transmission range (550 meters in our case).

In the mobility case, we study a random topology with two cases: low and high mobility. In both cases, it is only the node 1 that sends for the node n. The mobility model uses the random waypoint model [56], we justify our choice by the fact that the network is not designed for mobility and that this particular model is widely used in the literature. In this model the node mobility is typically random and all nodes are uniformly distributed in space simulation. The nodes move in 2200m*600m area, each one starts its movement from a random location to a random destination. Once the destination is reached, another random destination is targeted after a pause time.

B. Parameters Evaluations

We have simulated several scenarios with different numbers of nodes n and mobility values. We are interested in each scenario into two parameters. The first is the throughput which is given by the ratio of the received data on all data sent. The second parameter is the end-to-end delay which is given by (time for receipt of data - the data transmission time) / number of data packets received.

C. Simulations end Résults

In these scenarios, we compare our solution (IB-MAC) with MAC standard and two other solutions proposed in the literature. As like our IB-MAC improvement, these solutions use the MAC layer to improve TCP performance in the MANET. The first solution is Wireless Congestion Control Protocol (WCCP) [47] and the second one is MAC-layer LDA (Loss Differentiation Algorithm) [48]. The principle of each solution is given in the section of related work. Two cases are also considered, with and without mobility. In the case without mobility, we just use the chain topology since in our previous work [55] we have studied different topologies and we have found that it influence the communication environment in the same way. In both cases (with and without mobility), we used TCP New Reno version and AODV routing protocol. This choice is because the previous work [55] has shown almost similar results for different routing protocols and TCP versions used.

Scenario 1: Without mobility (Chain Topology), TCP New Reno, AODV.
Figure 2. End-To-End Delay variation without mobility (chain topology).

We see, through Figure 1, with MAC protocol, more the nodes number participating in the network increases, more the throughput decreases. This degradation at a given time (from n=100 nodes) begins to take stability. This degradation is due to TCP packet loss occurred, and that becomes more important with increasing size of the network. With the analysis of the trace files for these graphs, we found that RTS and CTS frames, handled at the MAC level, are sensitive to the network size, more the nodes number increases, the lose of these two frames increases too. It has been shown previously that such frames losses in such conditions of simulations are mainly due to the consequences of hidden and exposed nodes, a result that has already been achieved in our past work [15][16].

But when the IB-MAC is used as MAC protocol we see that the throughput is better. There is an important improvement of this parameter, even if there is a slight decrease when the number of nodes increases but this decrease is much smaller compared to the first case when the MAC protocol is used. This improvement is due to the use of the adaptive nature of our solution IB-MAC to the nodes number in the network.

Figure 2. shows the evolution of the second parameter studied which is the end-to-end delay when the nodes number increases. With MAC protocol, we find that this parameter significantly increases with the increase of the used nodes number. The increase of the end-to-end delay is essentially due to the detection of frequent loss of TCP packets in the network more the number of nodes increases. These losses will be the cause for the frequent start of the congestion avoidance mechanism by the TCP protocol, so that will result in delaying the transmission of TCP packets and the increase in delay. This increase in delay begins to stabilize from n = 110 nodes and that below t = 1.2 s approximately.

When the IB-MAC is used as MAC protocol we see that the end-to-end delay is better. There is an important improvement of this parameter, even if there is a slight increase when the number of nodes increases but this decrease is smaller compared to the first case when the MAC protocol is used.

Figure 1 and Figure 2 have been shown that our IB-MAC outperform not only MAC standard, but also similar techniques that have been proposed in the literature. The results of the variation of the throughput and the end-to-end delay parameters are better than those of MAC-LDA, MAC-WCCP and MAC standard.

The improvement of the throughput and end-to-end delay parameters is due to the dynamic nature of our new IB-MAC algorithm which makes the size of the backoff interval adjustable to the nodes number in the network. This adjustment reduces the probability of collisions between nodes, thus the number of lost packages is reduced while the throughput and delay are improved.

**Scenario 2:** Weak mobility (W = 5 m/s), TCP New Reno, AODV.
i) The increase of RTS/CTS frames losses with the increase of nodes number in the network (same to the first case without mobility);

ii) There are TCP packets losses even if there are successful RTS/CTS frames transmissions. In this case, these losses are caused by the unavailability route due the nodes mobility (the used route is outdated, denoted by "NRTE" in the trace file).

We deduce through i) and ii) that the mobility of nodes, although it is weak (here speed \(W = 5\) m/s), participates to the degradation of the throughput and end-to-end delay parameters.

With our IB-MAC solution, always with weak mobility, we found an important improvement of the throughput and end-to-end delay parameters in comparison to the first case when the MAC protocol is used. Our IB-MAC algorithm makes the size of the backoff interval adjustable to the nodes number in the network and their mobility. For this reason, even for the case where the nodes are mobiles, the probability of collisions between nodes is reduced, and then throughput and the end-to-end delay parameters are improved.

Figure 3 and Figure 4 shows also that IB-MAC outperform the others protocols used (MAC-LDA and MAC-WCCP). The results of the variation of the throughput and the end-to-end delay parameters are better than those of the others. Although there is a slight difference in performance but our strategy remains the best to the other three used here.

Scenario 3: Strong mobility (\(W = 25\) m/s), TCP New Reno, AODV.

For strong mobility (Figure 5 and Figure 6), we see that there is also a degradation of the throughput and end-to-end parameters when the MAC protocol is used, more important than the case with weak mobility because here the breaks connectivity increases then the links stability becomes more important. We have done the same analysis as above to know the reasons of this degradation; we found that the causes of this degradation are also related to those discussed in i) and ii) in weak mobility case.

In fact, when the network has a weak mobility (nodes with low speeds), it presents a rather high stability; then links failure are less frequent than the case of a high mobility. Consequently, the fraction of data loss is smaller when for the case where nodes move at low speeds (strong mobility), and grows with the increase in their mobility.

In this case too (strong mobility), with our solution IB-MAC, we found an important improvement of the throughput and end-to-end delay parameters in comparison to the first case when the MAC protocol is used. We found also a better improvement of the throughput and end-to-end delay parameters in comparison to the others protocols used (MAC-LDA, MAC-WCCP).

From these results, we can say that even in the case of a random topology where nodes are mobile (a feature specific to MANET networks) the IB-MAC solution improves the performance of TCP.

V. CONCLUSION

Improving TCP performance over 802.11 MAC protocol in multi-hop mobile ad-hoc networks is truly a problem on interaction between two layers. In this paper, we proposed an improvement of TCP protocol performance (throughput and end-to-end delay) in MANET. Our solution is IB-MAC which is a new Backoff algorithm making dynamic the \(CW_{max}\) terminal in depending on the number of nodes used in the network and their mobility. We studied the effects of IB-MAC on TCP performance, we limited our studies on very important parameters in such networks which are throughput and end-to-end delay because they have great effects on the performance.
of TCP protocol. The results are enough good and showed that our algorithm can outperform not only MAC standard, but also similar techniques that have been proposed in the literature like MAC-LDA and MAC-WCCP.

We do not claim that our IB-MAC solution is the optimal backoff algorithm to improve TCP performance, but the achieved results are indeed encouraging, justifying further investigation on this direction. The continuation of our work will consist in looking for a complete cross-layer IB-MAC in order to adapt dynamically and in a coordinated way the MAC and the TCP parameters.

REFERENCES


