A Cross-layer Approach to Enhance TCP Fairness in Wireless Ad-hoc Networks

G. Boggia, P. Camarda, L. A. Gricco, T. Mastrocristino, and G. Tesoriere

DEE - Politecnico di Bari, Via E. Orabona, 4 - 70125 Bari (Italy)
Email: {g.boggia, camarda, a.gricco, t.mastrocristino, g.tesoriere}@poliba.it

Abstract—Wireless ad-hoc networks represent an emerging technology, well suited for providing temporary network connectivity where a wired infrastructure cannot be easily set up. The very popular 802.11 standard enables wireless ad-hoc networking, using the Distributed Coordination Function (DCF) for multiple access to the shared radio channel. Unfortunately, the interaction between TCP dynamics, driven by the Additive Increase Multiplicative Decrease (AIMD) paradigm, and DCF channel access rules, which are based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm, leads to an inefficient spatial channel usage. As a consequence, 802.11 ad-hoc networks provide an unfair service to TCP flows. In order to achieve a reasonable trade-off between throughput and fairness, this paper proposes a cross-layer algorithm that dynamically limits the number of in flight segments in a TCP connection by taking into account measurements of frame collision probability, collected at the MAC layer along the path. Performance of the proposed algorithm have been evaluated by ns-2 simulations; results have shown that the developed cross-layer scheme provides the same goodput, improves fairness in bandwidth sharing, and reduces segment retransmission ratios with respect to the standard TCP over 802.11 MAC.

I. INTRODUCTION

Wireless ad-hoc networks represent an emerging technology, well suited for providing temporary network connectivity where a wired infrastructure cannot be easily set up [1]. They are made by smart nodes, provided with Wireless Network Interfaces, able to automatically build-up a multi-hop network, using ad-hoc auto-configuration protocols [2]. The 802.11 standard [3] enables wireless ad-hoc networking, using the Distributed Coordination Function (DCF) for multiple access to the shared radio channel. However, due to subtle interactions between TCP congestion control dynamics [4] and access rules of DCF, which adopts the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm [5], 802.11 ad-hoc networks provide an unfair service to TCP flows [1], [6]–[8]. This is a serious limitation, given that TCP carries a very large quota of the overall Internet traffic [9].

The reason of this behavior is related to the TCP congestion control which has been designed for wired networks: it additively increases the sending rate of TCP sources to probe the unknown network available bandwidth, until a congestion episode happens; the sending rate is multiplicatively reduced in response to a congestion to avoid network collapse [4]. In this way, during the probing phase, TCP allows a large number of segments to be outstanding, i.e., sent but not yet acknowledged. This, in turn, generates a high frame collision probability when TCP flows go through a 802.11 wireless ad-hoc network. As a consequence, a lot of frames are retransmitted and the throughput of TCP connections sharing the ad-hoc network is seriously decreased. Throughput degradation is more accentuated for those connections going through paths with a higher level of channel contention, i.e., paths involving nodes able to hear signals generated from many other wireless nodes in their neighborhood. In other words, in ad-hoc networks this behavior leads to an unfair service provision [1], [7], [8], [10], [11].

This paper proposes a cross-layer algorithm that, exploiting proper interactions between the MAC and the transport layers, dynamically limits the number of in flight segments in a TCP connection. Thus, a reasonable trade-off between throughput and fairness is achieved. The developed cross-layer algorithm acts on TCP flow control and requires only minor modifications to its classical implementation.

The rest of the paper is organized as follows: Section II briefly summarizes related work; Section III describes the proposed algorithm; Section IV shows simulation results; finally, the last Section draws the conclusions.

II. RELATED WORK

TCP behavior in 802.11 ad-hoc networks has been deeply investigated and new MAC and Transport protocols have been proposed to obtain performance improvement [1], [8], [10]–[16]. These studies have highlighted, as discussed in the previous section, that one of the main limitations to TCP performance in 802.11 ad-hoc networks is an inefficient spatial channel usage due to the interactions between TCP congestion control dynamics [4] and access rules of DCF [5]. As a consequence, 802.11 ad-hoc networks provide an unfair service to TCP flows.

It is widely demonstrated that this problem can be solved by properly limiting TCP transmission window, i.e., by exploiting some algorithms able to tune TCP window size as a function of the network load and of the channel contention level along the path [10]. For example, it has been shown that, in a $h$-hop chain topology, the optimal TCP window size is a fraction of $h$ [8], [10]. Finding an algorithm able to address the aforementioned issues is a hot research area.

In [12] the Neighborhood Random Early detection (NRED) algorithm has been proposed to enhance TCP fairness in
bandwidth sharing in 802.11 ad-hoc networks. In the NRED a node and its interfering nodes form a neighborhood to which a distributed queue made by the local queues of each node of the neighborhood is associated. NRED estimates the aggregate queue of each neighborhood and whenever this queue exceeds a given threshold, a drop probability is computed in accordance with the original RED scheme [17].

In [8] the Link-layer RED (LRED) is proposed. For each wireless node, LRED estimates the channel contention and properly tunes link-layer dropping probability by taking into account the perceived load.

In [10] the problem of properly selecting the TCP window limit is turned into the one of identifying the Bandwidth Delay Product (BDP) of a path in a Wireless Ad-Hoc Network. In particular, in [10] it is shown that the BDP of a path cannot exceed \( N \), where \( N \) is the number of the hop of the path. Then, an algorithm for setting the window limit as a function of \( N \) in a chain topology is proposed.

In [13] a novel hybrid channel access scheme that combines sender-initiated and receiver-initiated collision-avoidance hand-shakes is proposed for multi-hop ad-hoc networks. With the proposed channel access scheme, a node may operate in two modes: the Sender-Initiated (SI) and the Receiver-Initiated (RI). In the SI mode, basic 802.11 access rules are used. In the RI mode, instead, a three-way receiver initiated collision-avoidance hand-shake is used. RI mode is triggered only when SI mode does not perform well. In this way the burden to contend for the radio channel is distributed among nodes according to the different degrees of contention they experience.

In [16] it is shown that, by locally monitoring the transmission queue length of network interface and MAC layer behavior at each node, an approximation of the degree to which the wireless medium is busy can be estimated. Thus, several uses of such congestion information are proposed. In particular, regarding the goal of TCP performance optimization, the paper [16] proposes to set the Congestion Experienced (CE) codepoint into the IP header when the suggested congestion metric exceeds a certain threshold. When a TCP sender receives a packet with the CE codepoint set in its IP header, it responds as it would be in case of segment drop.

### III. The Cross-Layer Algorithm

The basic idea of our approach is to exploit the advertised window field \( \text{adw} \) of TCP segments to limit the transmission rate of the TCP sender\(^1\). As well known [18], the TCP sender cannot have a number of outstanding segments larger than the \( \text{adw} \) value advertised by its own receiver. Currently, the TCP receiver stamps into the \( \text{adw} \) field the space available into its receiving buffer, in order to avoid saturation by a fast connection. We propose to extend the use of \( \text{adw} \) in order to allow the receiver to limit the transmission rate of the TCP sender also when the path used by the connection exhibits a high frame collision probability. In particular, in the proposed cross-layer approach measurements about frame collision probability along the path are collected at the MAC layer and are communicated to the TCP receiver in order to properly set \( \text{adw} \).

To this aim, we assume the presence of a specific field, known as \text{nonCollisionProb} \( \text{field} \), in the MAC Protocol Data Unit (MPDU). This field is set equal to the non-collision probability \( p_{nc} \) at link level. In particular, \( p_{nc} \) is set equal to one in frames transmitted at the first hop of a TCP connection. At intermediate nodes, instead, \( p_{nc} \) is set by taking into account the collision probability \( p_{c} \) estimated by the wireless nodes from the ratio between the number of retransmitted frames and the total number of transmitted frames. After a frame reception, each node, before the forwarding, sets the \( p_{nc} \) value as follows:

\[
p_{nc} \leftarrow p_{nc}^{\text{old}} \cdot (1 - p_{c}),
\]

where \( p_{nc} \) is the value of the \text{nonCollisionProb} field reported into the received frame. In this way, at the last hop, i.e., the destination node, assuming collisions hop-by-hop as independent events, the overall collision probability along the path can be estimated by

\[
p_{c}^{\text{tot}} = 1 - p_{nc}.
\]

The value \( p_{c}^{\text{tot}} \) is compared with a given threshold \( p_{\text{thresh}} \); the result of the comparison is passed to TCP layer for setting the \( \text{adw} \) of the next ACK.

When \( p_{c}^{\text{tot}} > p_{\text{thresh}} \), i.e., the path exhibits a high collision probability, \( \text{adw} \) is decreased:

\[
\text{adw} \leftarrow \text{adw}_{\text{old}} - \text{adw}_{\text{old}} / \alpha,
\]

where \( \alpha \) is a parameter of the algorithm.

On the other hand, when \( p_{c}^{\text{tot}} \leq p_{\text{thresh}} \) for a number of consecutively received segments equal to the \text{threshCounter} value, \( \text{adw} \) is increased by one.

At the sender side, the \( \text{adw} \) of any new transmitted segment is copied from the analogous field of the last received ACK. In this way a control loop that inflates (reduces) \( \text{adw} \), when a low (high) frame collision probability along the path is detected, is established.

It follows the pseudo-code of the algorithm:

**TCP sender:**
- \( \text{adw} \) is set as the one of the last received ACK

**TCP receiver:**
- For each transmitted frame
  - \( p_{c} \) is estimated
- For each received frame
  - \( p_{nc}^{\text{old}} \) is read from the frame header
  - \( p_{nc}^{\text{old}} \cdot (1 - p_{c}) \) is stamped into the frame header

\(^1\)The algorithm has been conceived for unidirectional TCP flows, even if it could be easily extended to the bidirectional case with only some slight modifications.
- the frame is forwarded to the next hop.

First node of the path:
- \( p_{nc} \) is set equal to 1 in each transmitted frame.

IV. PERFORMANCE EVALUATION

We have analyzed the performance of the proposed algorithm using \( ns \)-2 simulations [19]. In particular, we have simulated the \( NxN \) grid scenario shown in Fig. 1.

![Fig. 1. \( NxN \) grid topology.](image)

Wireless nodes access the radio channel using basic DCF rules with a data rate of 11 Mbps. Three radio ranges have been defined: (1) the transmission range, set equal to 250 m, representing the maximum distance between two nodes for a meaningful frame exchange; (2) the carrier sense range, set equal to 550 m, being the radius within which a node can hear signals generated by other nodes; (3) the interference range, set equal to 550 m, defining the range within which nodes in receive mode will be interfered by other transmitting nodes. \( 2N \) greedy TCP connections are established across the grid topology and each connection goes through \( N - 1 \) hops. The New Reno TCP congestion control has been employed [6] with all TCP sinks implementing the delayed ACK option [20]. Segments are 1460 bytes long. The starting instants of the TCP connections are spaced by 0.1 s one to each other.

We have compared the performance of the proposed algorithm with the one of the standard TCP over 802.11 MAC, with \( N \) ranging from 3 to 15. For each value of \( N \), six simulations have been run by varying the seed of the \( ns \)-2 random number generator. Each run simulates the ad-hoc network for 300 s. We will report the average of the six results in order to filter out statistical fluctuations.

We will focus on TCP goodput, segment retransmission ratios, and fairness in bandwidth sharing. We have evaluated the performance of the proposed algorithm for many parameter sets; herein, we first report some significant results obtained for \( p_{thresh} = 0.05 \), \( \alpha = 7 \), and \( threshold_{counter} = 4 \), which provide a good tradeoff among all the considered performance indexes; then, we will show the parameter sensitivity of the algorithm.

Fig. 2 shows that, using the proposed algorithm, the total goodput obtained by the \( 2N \) TCP connections, for all the considered values of \( N \), is similar to that obtained using New Reno TCP over 802.11. In order to investigate how this goodput is distributed among the \( 2N \) TCP connections, we have evaluated the Jain fairness index [21]; this fairness index belongs to the interval \([0,1]\) and increases with fairness, reaching the maximum value at one. Fig. 3 shows that the adoption of the proposed cross-layer approach improves the fairness index for all values of \( N \). This means that the developed algorithm allows the bandwidth of the ad-hoc network to be fairly distributed among all the TCP connections. In fact, by limiting the number of outstanding segments into the wireless channel, a more efficient spatial usage of the ad-hoc network is achieved.

![Fig. 2. Goodput of the TCP connections.](image)

![Fig. 3. Fairness Index of the TCP connections.](image)
case, the reason is that by reducing the number of outstanding segments as soon as the collision probability increases, both frame collisions and network congestion episodes are reduced. This effect has significant impact on the energy consumption of wireless node which are usually battery supplied devices. Thus, the developed scheme can be fruitfully exploited when energy saving is an issue to be addressed [22].

In order to investigate the transient, Figs. 7-9 compare the goodputs, fairness indexes, and ratios of retransmitted segments obtained by the proposed algorithm and New Reno TCP over standard 802.11 as a function of the simulation time, still in the 5x5 grid. These figures show that the proposed algorithm provides basically the same goodput of New Reno TCP over 802.11 MAC, but strongly improves fairness and ratio of retransmitted segments also during the transient.

Finally, we look into the parameter sensitivity issue by reporting total TCP goodput and fairness index when a 5x5 grid is simulated using the proposed algorithm with many parameters sets. Figs. 10 and 11 shows that the $p_{thresh}$ parameter plays a major role, in fact when it is higher than 0.4 the fairness index exhibits unacceptable values, smaller than 0.7. The reason is that the larger $p_{thresh}$ is and less frequently the algorithm is triggered, with consequent degradation of the fairness index. Similar results have been obtained with other grids. From these results, it turns out that any value of $\alpha$ can be used provided that $p_{thresh} \leq 0.4$.

V. CONCLUSION

In this work a cross-layer approach has been exploited to optimize TCP performance in wireless ad hoc networks. The proposed cross-layer algorithm extends the use of the advertised window ($adw$) in order to allow a TCP receiver to slow down the respective TCP sender also when the path used by the connection exhibits a high frame collision probability. We have evaluated the performance of the proposed algorithm using ns-2 simulations, which have shown that the proposed cross-layer algorithm improves fairness in bandwidth sharing and provides smaller retransmission ratios with respect to standard TCP over 802.11 MAC.

REFERENCES

Fig. 8. Fairness Index of the TCP connections vs. simulation time.

Fig. 9. Retransmission ratio of the TCP connections vs. simulation time.

Fig. 10. Total TCP goodput in a 5x5 grid with several $\alpha$ and $p_{\text{thresh}}$ values.

Fig. 11. Fairness Index in a 5x5 grid with several $\alpha$ and $p_{\text{thresh}}$ values.


