SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks **00**: 1–15 (2010)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.0000

Characterizing the Greedy Behavior in Wireless Ad Hoc Networks

Soufiene Djahel[∓*], Farid Naït-abdesselam[∓] and Damla Turgut[±]

[∓]*LIFL – UMR CNRS USTL 8022 – IRCICA, University of Lille, France*
[±]*School of Electrical Engineering and Computer Science, University of Central Florida, USA*

Summary

While the problem of greedy behavior at the MAC layer has been widely explored in the context of wireless local area networks, its study for multi-hop wireless networks still almost unexplored and unexplained problem.
Indeed, in a wireless local area network, an access point mostly forwards packets sent by wireless nodes over the wired link. In this case, a greedy node can easily get more bandwidth share and starve all other associated contending nodes by manipulating intelligently MAC layer parameters. However, in wireless ad hoc environment, all packets are transmitted in a multi-hop fashion over wireless links. In this case, an attempting greedy node, if it behaves similarly as in a WLAN, trying to starve all its neighbors, then its next hop forwarder will be also prevented from forwarding its own traffic, which leads obviously to an end to end throughput collapse.
In this paper, we show that in order to have a more beneficial greedy behavior in wireless ad hoc network, a node must adopt a different approach than in WLAN to achieve a better performance of its own flows. Then, we present a new strategy to launch such a greedy attack in a proactive routing based wireless ad hoc network. A detailed description of the proposed strategy is provided along with its validation through extensive simulations. The obtained results show that a greedy node, applying the defined strategy, can gain more bandwidth than its neighbors and keep the end-to-end throughput of its own flows highly reasonable.

Keywords – QoS, Greedy Behavior, Ad Hoc Networks, Conflict Graphs.

1. Introduction

The increase in computation power, the compactness of size, incorporation of mobility and ease of connectivity from anywhere are amongst the major factors that resulted in tremendous growth of handheld devices in recent years. From cordless phones to cellular networks and from WiFi to sensors, the wireless medium has become the preferred backbone of today's deployed networks. The newest model being introduced is the Mobile Ad hoc Networks (MANETs), in which mobile nodes, within the transmission range of each others, can communicate directly over the wireless link, while those that are far apart use other nodes as relays. The properties of MANETs, such as shared wireless medium, open network architecture, stringent resource constraints and rapidly changing topology make this type of network vulnerable to a bunch of attacks at different layers, especially at MAC layer in which attacks are launched easily. Therefore, the task of securing

*Correspondence to: soufiene.djahel@lifl.fr

Prepared using secauth.cls *[Version: 2008/03/18 v1.00]*

such network remains hard and necessitates careful investigation.

Since IEEE 802.11 MAC protocol, as described in [5], is commonly used by wireless nodes to access the medium, any misbehavior at this level may affect the proper functioning of the network. The serious damage caused by MAC layer misbehavior has received considerable research attention leading to an in depth investigation and analysis of its root causes, such as the works done in [10], [9] and [21]. As a result of this investigation, some pioneering contributions have been proposed in the literature to cope with this problem such as [14], [16] and [17]. These works have identified several types of MAC layer misbehavior and proposed countermeasures to detect or prevent such misuse. However, their solutions are based on the assumption that the greedy node behaves similarly in MANETs as in WLAN. This assumption is neither realistic nor sustainable since the greedy node in MANETs, behaving exactly as in WLAN, will obviously not get much more advantages. Moreover, it may even disrupt the performance of its own traffic as it will be shown throughout this paper. Therefore, the existing solutions fail totaly in responding to the concern of greedy behaviors in MANETs.

In this paper, we show that in order to have a more beneficial greedy behavior in wireless ad hoc network, a node must adopt a different approach than in WLAN. This approach allows it to achieve better performance for its own traffic flow as well as for the crossing flows of interest. Then, we present a novel strategy to launch such a rational greedy attack in a proactive routing based wireless ad hoc network. A detailed description of the proposed strategy is provided along with its validation through extensive simulations. The obtained results show that a greedy node, applying our devised strategy, can gain more bandwidth than its neighbors and keep the end-to-end throughput and delay of its own flows highly reasonable.

The rest of the paper is organized as follows. The next section gives an overview on MAC layer vulnerabilities and provides a classification of greedy node's behaviors. Next, we give a brief overview on the literature followed by an in depth comparison between greedy behavior in WLAN and MANETs in section 4. our proposed greedy strategy in MANETs along with the assessment of the energy consumption induced are presented in section 5 and 6, respectively. In section 7, we report and discuss the obtained simulation results. Finally, section 8 concludes the paper.

2. Overview of MAC Layer Vulnerabilities

As it is well known, two medium access techniques exist in IEEE 802.11 MAC protocol, PCF (Point Coordination Function) and DCF (Distributed Coordination Function). While PCF is reserved for infrastructure based wireless networks (WLAN), the DCF technique can be used in both modes WLAN and infrastructureless based wireless networks such as MANETs, Mesh networks and vehicle to vehicle networks. Therefore, we discuss in the sequel the potential vulnerabilities of the DCF mode.

A misbehaving node may disobey the MAC protocol rules to gain more bandwidth over regularly behaving honest nodes. To do so, it should change the MAC layer parameters. A node can modify the MAC parameters configuration only if the network access card runs the WIFI protocol on software. In this case, the misbehaving node can easily implement the following misbehavior techniques:

- Selects its backoff values from different distributions, for example the backoff period is randomly picked out from the interval [0, $k \times CWmin$] where $0 \leq k \leq 1$. Note that if k = 1 then the cheater behaves correctly however it doesn't double its CW after a collision is occurred. Moreover, it can use different retransmission strategies upon experiencing an unsuccessful transmission.
- Jams the CTS or ACK frames of its neighbors in order to increase their contention windows.
- When the channel is sensed to be idle, it transmits before the required DIFS time slots elapses, i.e. the misbehaving node waits for a shorter period called S-DIFS (Short-DIFS). This misbehavior technique is significant only if the cheater node's backoff was already elapsed before it defers its transmission or if it sets its backoff value to zero.
- Amplify the value of the duration field in RTS or DATA packets such that the receivers keep silence for a period larger than the real transmission time. Consequently, if the cheater node has more packets to send, it gets more chance to access the medium as it starts counting down its backoff before its neighbors.

Greedy nodes' classification

The misbehaving nodes applying the above strategies can be classified according to the adopted strategy to launch the attack and the extent of the induced harm.

DOI: 10.1002/sec

Hence, the following classes of greedy nodes can be identified (see Fig. 1):

• **Indirect greedy node**: which aims to increases its bandwidth by launching a cross-layer attack targeting the routing protocols in order to decrease the number of contending nodes around it, and then increases its chance to frequently access the medium. To this end, it may either increases its SIFS value to cause RTS timeout at the sender node or deny response to the received RTS frames.

• **Direct greedy node**: which manipulates the medium access parameters such as backoff, DIFS or jams the CTS/ACK packets of its neighbors. This class can be further divided into three sub-classes as follows:

- **Malicious greedy node**: which aims to disrupt the ongoing communications in its neighborhood and cause damage to network performance without seeking for any benefits. It can even send fake data packets to monopolize the medium, under the assumption that it is equipped with a permanent energy supply.
- **Rational & Selfish (uncooperative) greedy node**: The greedy node here wants to increase the throughput for its own traffic flow and decrease its end-to-end delay. However, it affects the performance of the crossing flow by delaying it and releasing the medium for the next hop of its flow to forward the transmitted packets.
- **Rational & Cooperative greedy node**: In this case, the greedy node's behavior is similar to the previous category; however it chooses some crossing flows which are of direct interest or conveying critical information in order to favor them during forwarding. The following situations justify this behavior of the greedy node.
  - In a battlefield the orders issued from the group leader are critical and need to be prioritized than the other flows.
  - In an emergency area, the packets sent by the rescuers which are inside the area of incident are critical and should be given high level of importance by the forwarder node.

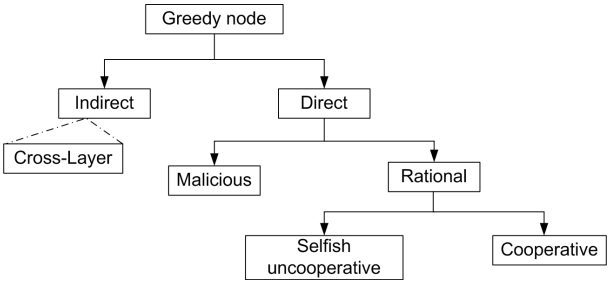

Figure 1. Classification of the greedy nodes's behaviors

3. Recent attempts to prevent/detect MAC layer misbehavior

In the last few years, several counter-measures have been proposed to protect the network against MAC layer misbehavior in both WLAN and MANETs environments. In the former environment, the majority of the proposed solutions takes advantage of the trustworthiness property of the access point and designs centralized schemes based on this property. In the latter environment, the aforementioned property is no longer valid that's why the researchers propose distributed solutions rather than centralized. Those solutions are either defining new MAC protocols completely different from the standard IEEE 802.11 or maintaining the DCF protocol unchanged and add new components to monitor the surrounding nodes and collect statistics about their behaviors or just adjust their own DCF parameters according to a specific game.

3.1. WLAN environment

The authors of [17] have presented a modular system, dubbed DOMINO which does not require any modification to the standard MAC protocol. This system is implemented at the AP (access point) which is assumed to be trustworthy. It consists of a set of components ensuring complementary tasks. The first task is to monitor the behavior of wireless nodes around the AP for a certain period of time in order to collect traffic traces of each node. As a second task, these traces are passed through a set of tests to measure the deviation of each node from the expected regular behavior. Each of these tests corresponds to a specific misbehavior technique (e.g. backoff manipulation, reducing DIFS value and jamming CTS frames). The output of these tests is analyzed by the decision component to infer whether a given node is well behaved or greedy. A node is

DOI: 10.1002/sec

considered as greedy if its corresponding deviation counter exceeds a predefined threshold for at least one test. The network administrator is then informed about the detected cheaters in order to punish them adequately.

DOMINO fails to detect an adaptive greedy node which alternates randomly among several misbehavior techniques in order to escape from the detection engine, however it still achieving a higher bandwidth. To circumvent this weakness of DOMINO, [23] proposes a novel countermeasure based on fuzzy logic dubbed FLSAC. The main idea of this scheme is to carry out a global estimation of node's deviation from the standard whenever this node is deemed as well behaved by DOMINO (i.e, double check). The aim of this verification is to check whether the combined deviations of a node with respect to the misbehavior techniques discussed earlier allow it to earn a considerable extra bandwidth. If so, this node is deemed as greedy and the same reaction as in DOMINO is applied.

The main advantages of those solutions is that the reaction or punishment of the detected cheaters is ensured by a trustworthy entity which is the AP. Furthermore, the task of disseminating the identity of the detected node is no longer needed.

3.2. Wireless multi-hop networks environment

The sequential analysis concept introduced by Wald in [1] was widely used by researchers to struggle security attacks in wireless networks. The scheme presented in [18] is based on this concept; it describes an analytical model for the packets inter-arrival time distribution in saturated networks, representing an extension of Bianchi 's stochastic model [6]. Based on this model the authors have developed an algorithm to detect the cheating nodes by observing the throughput earned by each node. These observations are further evaluated through a sequential probability ratio test to identify which node is not obeying the protocol rules. To ensure its correctness, this scheme assumes the knowledge of the exact value of the greedy factor (i.e., the interval from which the greedy node selects its back off value), however this information is not available in practice. Therefore this scheme cannot work in real environment.

In [20], a statistical framework is developed in order to detect selfish nodes which deliberately modify their contention window to increase their throughput. First, a sample of number of idle slots between the successful transmissions of each node is collected. Subsequently, the Kolmogorov-Smirnov (K-S) [2] test is applied to distinguish the misbehaving nodes (using unpredictable strategy) from the legitimate ones. Notice that two detectors have been proposed by the authors, a batch detector based on Neyman-Pearson test and q sequential detector based on Wald's test. The results have shown that both of the detectors successfully identify the cheaters for the majority of the applied strategies.

In order to guarantee a faster detection of the cheaters, the authors of [22] have developed the PRB (Predictable Random Backoff) algorithm. PRB is based on slight modification of the standard backoff algorithm by forcing each node to choose its backoff value from the interval [CW_{lb}, CW] instead of [0, CW], where CW_{lb} is calculated based on the previous backoff value and CW is a function of CW_{min} along with the number of failed transmissions. In this way, a receiver node can detect any deviation from the sender since the backoff value is predictable. This solution is faster than the previous ones however it presents the following drawbacks:

- The backoff value observed by the receiver may be different from the one generated by the sender due to hidden terminal phenomenon, interference and inter frame delay of TCP traffic. Hence per frame detection may increase the probability of triggering false alarms and consequently punishing honest nodes.
- Since in PRB each node selects its backoff from a smaller interval as compared to the BEB (Binary Exponential Back off) algorithm the number of collisions increases, leading to higher packet delay and low channel utilization.

Game theory has been widely applied for investigating and assessing the selfish behavior impact in CSMA/CA, and numerous contributions have been proposed to cope up with. In [15], the authors have proven that a selfish and uncooperative behavior of a small number of attackers results in harmful damage to the network. To prevent such situation, they have proposed a dynamic game based scheme and derived the conditions which lead the set of cheaters to reach the Pareto optimal Nash equilibrium. Furthermore, they have also proposed a detection mechanism for non cooperative cheaters along with an adequate punishment scheme.

The common disadvantages of the herein described solutions is that none of them provides an efficient reaction scheme upon correct identification of a greedy node. Furthermore, the revelation of the greedy

DOI: 10.1002/sec

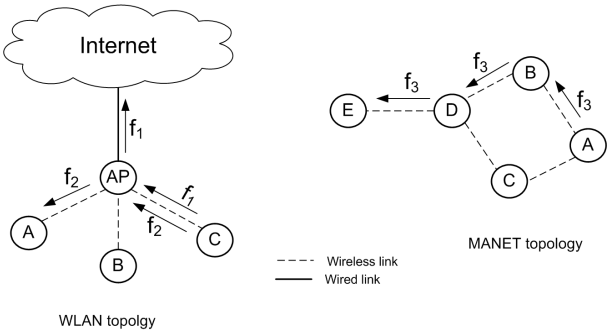

Figure 2. Greedy behavior: WLAN versus MANETs

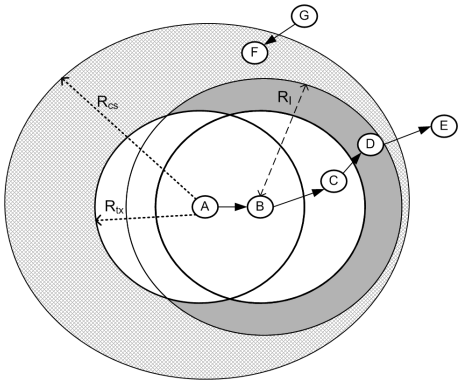

Figure 3. Propagation of greedy behavior's impact in MANETs

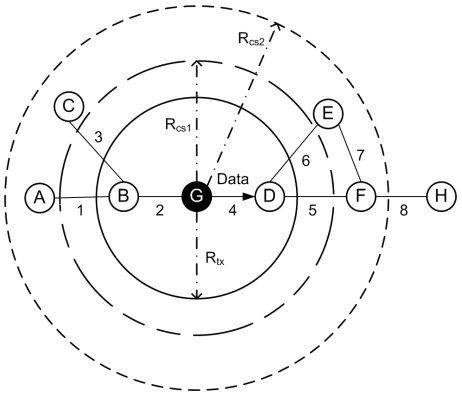

Figure 4. The connectivity graph

node's identity to the whole set of well-behaved nodes remains an open issue.

4. Greedy behavior impact on network performance: WLAN versus MANETs

In this section we emphasize the major difference between the greedy behavior in WLAN and MANETs. In other words, we try to answer the following question: Are the damages induced by greedy nodes in WLAN and MANETs similar?

As illustrated in Fig. 2, the destination of a flow in WLAN can be either a far away node or the one attached to the same access point (AP). In the former case, the source node of the flow f_1 tries to gain the entire bandwidth regardless of the decrease in its neighbor's throughput. This is due to the fact that its next hop (AP) forwards the packets of the flow f_1 through a wired link, independent from the wireless ones (no transmission conflict exist between those links). The flow f_2 is similar to the case of the flow f_3 in MANETs, any attempt of the flow's source node A or an intermediate node B to dominate the medium deprives its next hop from forwarding the received packets. Consequently, the flow's performance collapses sharply. Furthermore, the impact of this misbehavior may propagate to affect other flows crossing through the nodes in contention with the greedy node.

To illustrate this phenomenon related to radio wave propagation, let us consider the network topology given in Fig. 3. In this figure, R_{tx} and R_{cs} represent the transmission and carrier sensing ranges of node A, respectively. The lightly shaded area represents the region which is not covered by RTS/CTS handshake between A and B. Note that any transmission initiated from a node within this region may not interfere with packet reception at node B as these nodes are out of its interference area, represented by the darker region which is delimited by the interference range R_I. Despite that, the nodes within the lightly shaded area have to differ their transmissions since they sense the medium busy due to node A's transmission. As a result, if the sender node A misbehaves and monopolizes the medium for a long duration, all the transmissions over the links where at least one node is within the lightly shaded area are delayed leading to an increase on the number of dropped packets and the end-to-end delay. Even the links (B,C) and (C,D) are negatively affected meaning that the greedy node A is increasing its throughput in the detriment of the quality of service requirements of its own traffic flow.

On the contrary of MANETs, the situation discussed above does not arise in WLAN environment since all the nodes are within the transmission range

DOI: 10.1002/sec

of the AP, therefore the increase of the greedy node's throughput does not affect the end-to-end delay of its traffic flow. As a conclusion, for a more effective greedy behavior the greedy node should choose an alternative strategy adapted to the constraints of MANETs environment.

Illustrative example The Fig. 5 shows an example of the medium access frequency by a greedy node sending a traffic flow, its next hop node and the other neighbors in case where this greedy node tries to dominate the wireless medium. Therefore, this behavior leads to starving the greedy node's neighbors, including its next hop node, from retransmitting the received packets. If we consider the simple case where the destination node is two hops away from the sender (greedy) then the end-to-end delay (Ed_i) is computed as

$$Ed_i = 2T_{pi} + \sum_{j=i+1}^{x} (T_{pj} + CT_j) + \sum_{j=1}^{i-1} (T_{pj} + CT_j) + T_1 \quad (1)$$

where T_{pi} denotes the one hop transmission time of the packet pi, CT_j refers to the contention time spent by the node before accessing the medium and T_1 is the period during which the node is differing as one of its neighbors is transmitting. In the case where the flow's source node is behaving correctly the end-to-end delay Ed'_i is expressed as

$$Ed'_i = 2T_{pi} + \sum_{j=1}^{i-1} (T_{pj} + CT_j) + T_1 \quad (2)$$

then

$$Ed'_i - Ed_i = \sum_{j=i+1}^{x} (T_{pj} + CT_j) \quad (3)$$

This extra delay (Ed'_i - Ed_i) increases sharply as the number of packets (x) to be transmitted by the greedy node gets higher leading to devastating impact on the traffic flow performance and violates all the QoS requirements.

5. Designing new greedy strategy for MANETs

In this section, we give the road map of the required steps for the greedy node to launch the greedy attack according to our strategy. First, we provide the basic assumptions of our scheme followed by a description of how the greedy node constructs the conflict graph. Next, we show how to extract the bandwidth fair share of a node according to the conflict graph. Afterwards, we determine the maximum extra bandwidth the greedy node can gain without negatively affecting its traffic flow performance. Finally, we present the algorithm used by the greedy node to launch the greedy attack and to ensure the accordance with the values computed in the previous step.

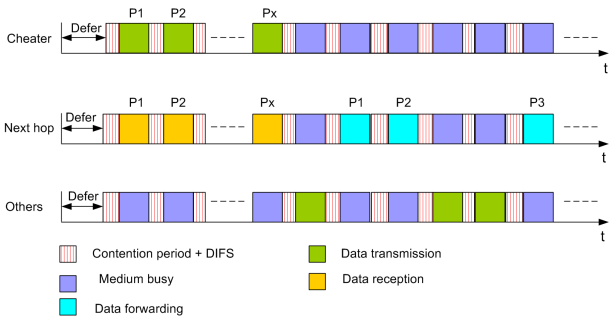

Figure 5. Example of bandwidth share among the greedy node, its next hop and the other neighbors nodes in the case where this greedy node is applying full greedy strategy (similar to WLAN case) in order to monopolize the medium

5.1. Main Assumptions

We give an overview of the assumptions used throughout the paper. These assumptions constitute the core of our cheating strategy.

- A proactive routing protocol is used at the network layer to establish end-to-end routes such as OLSR †[12] .
- Carrier sensing range (R_{cs}) is equal to more than twice of the transmission range (R_{tx}) [4], whereas the signal propagation follows the 2-Ray Ground Reflection Model ‡ [24].
- The nodes are distributed within the topology according to the Poisson process of parameter λ [3].
- We assume CBR traffic with fixed packets size S and the transmission rate offered by the underlying MAC protocol is β. Therefore,

†Notice that we can use any other routing protocol which provides the same topology view as OLSR.
‡The two-ray ground model is a common propagation model that has been widely used in wireless communications. Applying different propagation models could change the result, but the change is expected to be subtle.

DOI: 10.1002/sec

the number of time slots (η) needed for the transmission of the packet payload is:

$$\gamma = \frac{(\frac{S}{\beta})}{\eta} \quad (4)$$

- The length of a packet P_l is defined as the number of time slots required for its successful transmission and can be expressed as follows:

$$\begin{aligned} P_l &= \frac{(NAV - T_{DATA}) + DIFS + H_l}{\eta} + \gamma \\ P_l &= \frac{Duration + DIFS + H_l}{\eta} + \gamma \quad (5) \end{aligned}$$

where

$$Duration = T_{RTS} + T_{CTS} + T_{ACK} + 3 \times SIFS$$

Notice that T_{RTS}, T_{CTS} and T_{ACK} refer to the transmission time of RTS, CTS and ACK frames respectively, whereas H_l denotes the aggregate length of Physical, MAC and UDP headers defined as follows:

$$H_l = \frac{PHS + MHS + UHS}{\beta} \quad (6)$$

where `PHS`, `MHS` and `UHS` are the size of PHY, MAC and UDP headers respectively.

5.2. Conflict graph construction

As a first step of our scheme, the greedy node constructs the contention flow graph with nodes within its R_{cs} to derive its predicted fair-share of bandwidth [13]. To this end, the greedy node analyzes the received information in Hello and topology control (TC) messages and constructs its conflict graph [11] accordingly. For example, node G in the topology shown in Fig. 4 acquires the set of its 2-hops neighbors A, C, E and F from the Hello messages sent by nodes B and D, and it discovers its 3-hops neighbor H from the TC message sent by the node F which is multipoint Relay (MPR) of node H.

After acquiring the necessary information, the greedy node G constructs the conflict graph within its carrier sensing range, from which it extracts the set of maximal cliques. Since the topology information acquired from Hello and TC messages is partial, node G constructs this graph by considering the worst case scenario assuming the maximum number of contending links to compute the minimum bandwidth fair share. The number of maximal cliques is the key for determining the misbehaving threshold which will be discussed later. As shown in Fig. 6, the conflict graph depends on the extent of the carrier sensing range of the greedy node, for which we distinguish two cases:

- R_{cs} is slightly larger than the transmission range R_{tx} (see Fig. 4, R_{cs} = R_{cs1}), thus we have less contention between links and consequently the greedy behavior impact reduces. According to the set of maximal cliques shown in Fig. 7(a), only a simultaneous transmission over the following pair of links is allowed:

(1,5)	(3,5)	(2,7)
(1,6)	(3,6)	(2,8)
(1,7)	(3,7)	
(1,8)	(3,8)	

- R_{cs} is greater than twice of the transmission range,
  $R_{cs} > 2 \times R_{tx}$ (see Fig. 4, R_{cs} = R_{cs2}) which means that all the 2-hops neighbors of the greedy node G are within its carrier sensing range. Hence, only few links can be active for flow transmission at the same time as depicted in Fig. 7(b) where only the pairs of links $(1,7)$, $(1,8)$, $(3,7)$ and $(3,8)$ are allowed to transport traffic flows simultaneously. As compared to the first case, the number of conflict between links raises leading to devastating consequences if one node doesn't obey the MAC protocol rules.

Time complexity for generating the maximal cliques . Given that for N nodes we have at most $\frac{N(N-1)}{2}$ links which can be established between them. According to the algorithm by Tomita et al. [19], the worst-case time complexity for generating the set of maximal cliques from the graph constructed by those links is estimated to $O(\ 3^{N/3})$.

5.3. Bandwidth fair-share estimation

Once the conflict graph is established and the set of maximal cliques is derived, the node G computes its

DOI: 10.1002/sec

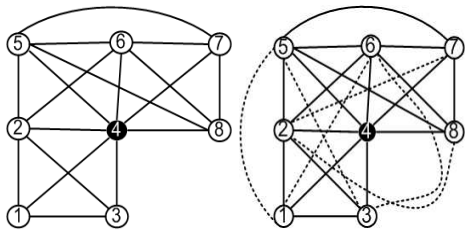

Figure 6. Conflict graph of the contending transmissions. (a) case of R_{cs} = R_{cs1}; (b) case of R_{cs} = R_{cs2}

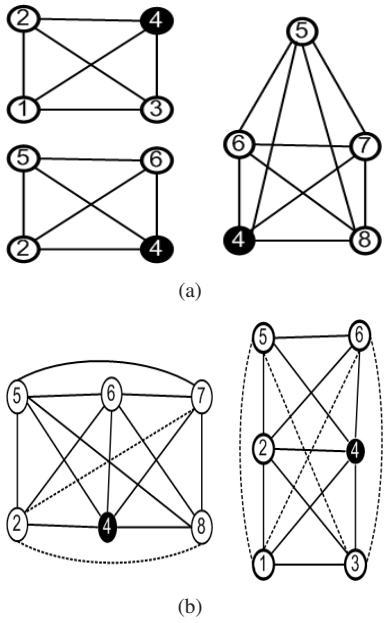

Figure 7. The set of maximal cliques. (a) R_{cs} = R_{cs1}, 3 maximal cliques whose sizes are 4, 4 and 5 respectively; (b) R_{cs} = R_{cs2}, 2 maximal cliques of 6 vertices each. Note that the dashed edges represent the new links created due to increase in R_{cs} from R_{cs1} to R_{cs2}

fair share of bandwidth and the end-to-end throughput of its traffic flow. In order to compute these values, we assume each node has a nonempty buffer of packets ready to be transmitted at each time slot (saturation case). Hence, given a particular path relaying source and destination nodes, the end-to-end throughput capacity is defined as the minimum link throughput capacity B_i of this path. As the links in conflict with the link 4 ($G \rightarrow D$) are the bottleneck for any flow crossing them when G is cheating, we determine the end-to-end throughput E_{th} as the minimum capacity of these links. This means if the length of any path from source to destination is n hops, the end-to-end throughput is computed only for the m hops ($m \leq n$) within the sensing range of the source node, which can be formulated as

$$B_i = sc_t \times (1-\tau)^{\lambda\pi R_{cs}^2 - 1} \times \beta \times \frac{\gamma}{P_l} \quad (7)$$

$$E_{th} = min(B_1, ..., B_{\lambda\pi R_{cs}^2}) \quad (8)$$

where sc_t denotes the duration of successful and collided transmissions of node i and τ is the probability of transmission. For more details on the computation of the values above, the reader may refer to the work by Wang and Garcia-Luna-Aceves [8].

Since we construct a partial topology graph limited to links in which at least one of the extremity is within the carrier sensing range of node G, then the calculated fair-share B_i of a node i might be greater than the real one R_i. That is the reason why R_i can be expressed in function of Bi as

$$R_i = B_i \times \Phi \times \Psi \quad (9)$$

such that Φ is a factor used to adjust the estimated faire-share to the real one, where

$$0.5 \leq \Phi \leq 1$$

Moreover, as the greedy node constructs the topology graph by considering the maximum number of links relaying its 2-hops neighbors and between theses nodes and its three hops neighbors, the computation of the fair-share is done based on links which might not exist. For that reason, the value Ψ, dubbed density factor, is used to increase this fair-share accordingly. This value is determined upon the following criterions:

- Nodes' density in the neighborhood of the greedy node.
- The number of MPR nodes selected by the greedy node; if this number is small and the density of nodes within its carrier sensing range is high then it is more likely to have more links between nodes, and consequently Ψ can be assigned a value close to 1. On the other hand, if the MPR set is large and the nodes' density is mediocre then the value of Ψ can be increased more than the previous case.

The value Ψ is expressed as

DOI: 10.1002/sec

$$\Psi = 1 + \frac{|MPR\_set|}{|S_1 \cup S_2|} \quad (10)$$

where S_1 and S_2 denote the sets of node G's 1- and 2-hops neighbors respectively.

Remark In our proposed strategy, one may argue that a node can request its one or 2-hops neighbors for exchanging topology information in order to get the complete view of the network. However, such an action makes the node suspicious which may facilitate its detection if any anti MAC layer misbehavior system is deployed. Moreover, none of its neighbors will respond to these requests since they are not considered proper operations of the routing protocols. As a consequence, our proposed algorithm is more secure and realistic since it depends only on the information gathered locally by the greedy node from the legitimate exchange of control packets.

5.4. Misbehaving Threshold Computation

In this section, we define an upper bound of the extra bandwidth earned by the greedy node, dubbed misbehaving threshold. Any greedy node overtaking this threshold will experience a decrease of its own flows' performance (in terms of end-to-end throughput and delay).

As known, in the case where fair-share is held amongst the contending nodes, the greedy node gets R_i of bandwidth. When the greedy node misbehaves, its share is R_i + B_g which means that it acquires B_g of extra bandwidth share as a result of its mischief. So, for a rational greedy node, the value B_g should satisfy the following condition

$$\begin{aligned} \frac{[B - (R_i + B_g)]}{(N-1)} &> \alpha \times E_{th} \\ (B - R_i - B_g) &> (N-1)\ \alpha \times E_{th} \end{aligned}$$

therefore

$$B_g \leq B - R_i - (N-1)\ \alpha \times E_{th} \quad (11)$$

such that, N is the average number of nodes within the carrier sensing range, R_{cs}, of the greedy node, which can be expressed as
$N = \lambda \pi R_{cs}^2$. B is the total bandwidth available and E_{th} is the estimated end-to-end throughput of the ongoing flow calculated according to the formula given in Eq. 8.

The reason of using the condition above is the fact that any adopted misbehaving strategy which reduces the mean [§] of the greedy node neighbors' throughput below the value of E_{th} has also a negative impact on its own flow's performance. Hence, the rational greedy node has to ensure that B_g fulfils the condition given in Eq. 11 in order to satisfy the QoS requirements of its flow.

Notice that the value α is used to adjust the extra bandwidth gain of the greedy node with respect to the topology of the bottleneck area (the area covered by R_{cs}) and the contention flow graph. It can be expressed as

$$\alpha = \frac{\sum_{i=1}^{N} MC_i}{N \times cl} \quad (12)$$

where MC_i denotes the number of maximal cliques to whom the link $i \leftrightarrow j$ belong such that the node i is either sender or receiver over this link, and cl is the total number of the maximal cliques in the conflict graph.

5.5. Launching the rational greedy strategy

Once all the parameters defined in the previous steps are computed, the greedy node G carries out the two misbehavior techniques described below to achieve its goal.

It first selects a small Backoff value in order to gain more bandwidth within its allowed threshold. Then, it provokes collisions with its neighbors' frames except the frames of its ongoing flow's next hop by simply scheduling a transmission of a small or empty packet whenever it receives an RTS which is not destined to it and not sent by its next hop node. This process is illustrated by the Algorithm 1.

These two steps needs to be adjusted according to the misbehaving threshold, B_g, which means that the greedy node must compute the bandwidth share acquired by its next hop and adjusts its jamming rate and contention window accordingly. This process is described in Algorithm 2. In this algorithm, the estimation of the next hop's bandwidth is periodically computed whenever the timer period is expired.

[§]We use the mean of throughput of the greedy node's neighbors as there is a common bandwidth fair-share for each of them.

DOI: 10.1002/sec

Algorithm 1 Greedy node behavior

```
if (RTS received) then
    attempt = false;
    if (@Dest == my address) then
        schedule CTS transmission;
    else
        if ( @source ∉ NH-set ) then
            if (++CPT < n1) then
                schedule transmission of empty or small packet after SIFS;
            end
            CPT = CPT mod n2;
        else
            attempt = true;
        end
    end
end
```

/*where NH-set is the set of the greedy node's next hops for all the flows. n_1, n_2 and CPT are values used to adjust the jamming rate such that $n_1 < n_2$. */

Algorithm 2 Next hop bandwidth estimation and adjustment of the cheating parameters accordingly

```
if ((DATA received) && (attempt == true) then
    B_NH = B_NH + P_l;
    if (Period elapsed) then
        if (B_NH < E_th ) then
            increase jam rate;
            if (B_own > R_share + B_g) then
                decrease k;
            end
        else
            if ((E_th - B_NH) > threshold) then
                decrease jam rate;
            end
        end
        B_NH = 0;
        B_own = 0;
    end
end
```

/* B_{NH} and B_{own} are the bandwidth gained by the next hop and the greedy node respectively, expressed in number of time slots, during each period. The value $Threshold$ is used by the greedy node to prevent wasting more energy in jamming whenever its goal is achieved. k is the misbehavior coefficient used to choose a small backoff value as described in section 2. */

6. Energy Constraints

The mobile nodes in ad hoc networks usually need to be autonomous and independent from any central fixed infrastructure, and thus powered by batteries providing limited energy supply. In order to establish routes towards far away destinations, each node have to participate in a distributed routing protocol by exchanging broadcast/unicast control packets, leading it to spend more energy. Since our proposed greedy strategy is based on a proactive routing protocol, known by its heavy control traffic, so an important part of the energy is consumed in sending and receiving this traffic. Therefore, for energy awareness perspectives, the greedy node needs to minimize the energy wasted in jamming the frames sent in its neighborhood, otherwise its energy depletes rapidly. In order to minimize the consumed energy a periodic tuning of the jamming rate is applied as described in the Algorithm 2. This algorithm shows that the greedy node jams its neighbors' CTS frames at a minimum rate in order to allow its next hop of the ongoing flow to achieve the appropriate throughput.

According to the study done in [7], the energy consumed by the network interface for sending, receiving or discarding a packet is expressed as a linear equation:

$$cost = m \times size + b \tag{13}$$

such that the linear coefficients m and b represent an incremental cost proportional to the packet size and the cost of medium acquisition, respectively. In our strategy, the cost of jamming one CTS packet is given as follows:

$$\begin{aligned} cost_{jam} &= m_{jam} \times size_{jam} + b \\ &+ \sum_{|S_1|} (cost_{recv} + cost_{disc}) \end{aligned} \tag{14}$$

where $cost_{recv}$ and $cost_{disc}$ reflect the cost of the reception of the small packet sent by the greedy node and its destruction by its neighbors, respectively. This emphasizes the importance of the proposed strategy in terms of minimizing the jamming rate (i.e. jams only if necessary) in order to provide a QoS guarantee for the running application.

To evaluate the amount of energy wasted for running our scheme, let's assume that during one second of network lifetime X RTS frames have been successfully sent and the greedy node has provoked collisions with $X \times jam\_rate$ CTS frames, where $0 \leq jam\_rate < 1$. Therefore, the overall energy $E_{overall}$ consumed by the greedy node for jamming others' frames during the network lifetime T can be expressed as

$$E_{overall} = T \times X \times jam\_rate \times cost_{jam1} \tag{15}$$

where
$$cost_{jam1} = cost_{jam} - \sum_{|S_1|} (cost_{recv} + cost_{disc})$$

DOI: 10.1002/sec

Parameters	Values
Area	2000m × 1000m
Physical layer	direct sequence
Transmission range	250m
Carrier sensing range	550 m
Traffic type	CBR
Data rate	2 mbps
CBR packets size	500 bytes
Buffer size	64 packets
Simulation time	300 seconds
# simulation epochs	5
Network simulator	OPNET 14.0 [25]

Table I. Simulation settings

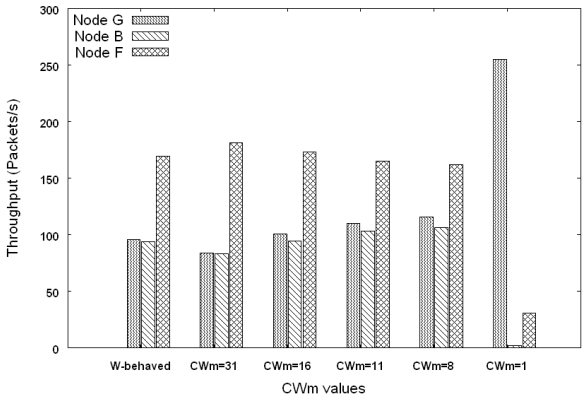

Figure 8. Propagation of greedy behavior's impact according to CW_m variation in MANETs, measured in terms of the acquired throughput.

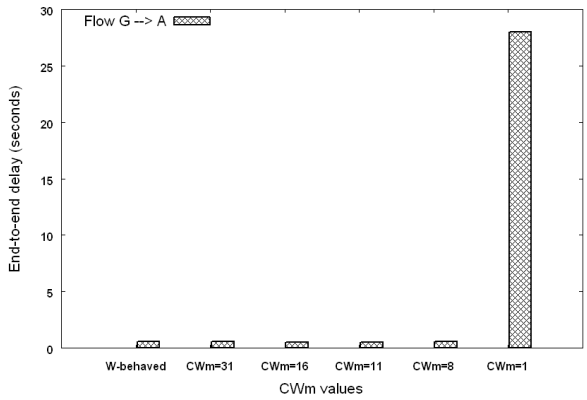

Figure 9. End-to-end delay of the greedy node's flow versus CW_m size.

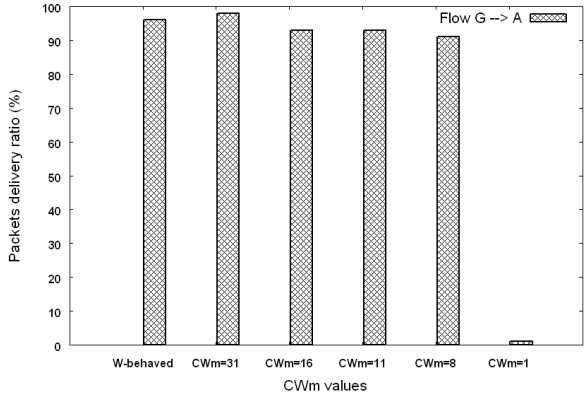

Figure 10. Variation of the packet delivery ratio of the greedy node's flow versus the chosen CW_m value

7. Experimental study

We now proceed to the experimental evaluation of our proposed greedy strategy. First, we illustrate the propagation of the greedy behavior's impact in ad hoc networks. Then, we emphasize the benefits gained by the greedy node in terms of throughput, end-to-end delay, and delivery ratio of its traffic flow when it behaves according to our strategy. The simulation parameters are summarized in Table I.

7.1. Propagation of Greedy Behavior Impact

In our experiments, we consider the same topology shown in Fig. 4 where two traffic flows are generated, G → A and F → H. The traffic sources send 1000 bytes every 2 ms (500 packets/s each) which means each source node has a packet ready for transmission at each time slot. Fig. 8 plots the obtained throughput by the greedy node G, its next hop node B and the node F with different values of the contention window of node G. When node G behaves correctly or sets its contention window constantly to 31 (equivalent to the minimum contention window CW_m), node F gets more bandwidth since it has less contention than node G. As we can see from the network topology, the location of node F favors it to seize the channel more likely than nodes G and B, leading to short term unfairness as well as long term unfairness. For example, during node B's transmission node F monopolizes the channel by transmitting continuously over the link 8 (all links in conflict with this link are inactive) and then increases its chance to transmit before node G which is deferring due to node B's transmission.

The throughput earned by the node F decreases slightly with the decrease of node G's contention

DOI: 10.1002/sec

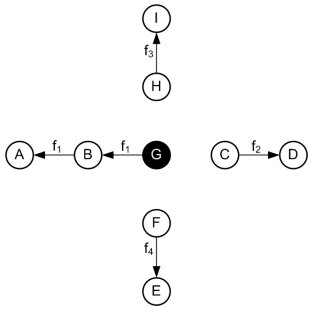

Figure 11. Topology used for evaluation of our proposed greedy behavior strategy.

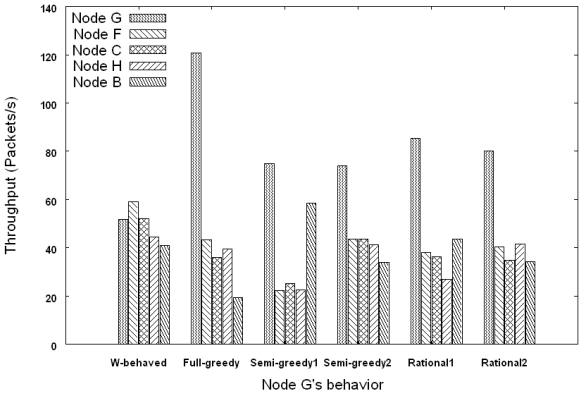

Figure 12. Variation of the traffic flows sources' throughput with the different cheating strategies adopted by the greedy node G.

window, whereas the throughput of nodes G and B is increasing, until it collapses sharply when node G's contention window is set to 1. When node G monopolizes the medium by choosing constantly a backoff value equal to 0, CW_m = 1, the throughput of its next hop, node B, drops sharply and consequently the delivery ratio of the flow G $\rightarrow$ A drops as well (see Fig. 10.)

From Figures 8, 9 and 10 we note that the misbehavior of the node G has a devastating impact only when it constantly sets its CW_m to 1 where the end-to-end delay for the small portion of packets forwarded by node B becomes quite long leading to the violation of the running application's QoS requirements. Moreover, this impact propagates to affect any other traffic flow within nodes G's carrier sensing range which makes this area a bottleneck in the network.

7.2. Advantages of the proposed greedy behavior strategy

In this section, we highlight the advantage of adopting our strategy by the greedy node. We consider the topology shown in Fig. 11, where four traffic flows f_1, f_2, f_3 and f_4 are generated in the network such that each source node sends 200 packets per second of 500 bytes each. In this scenario, we vary the node G's behavior among different strategies and observe the impact of each on the throughput, end-to-end delay and the packet delivery ratio. From the simulation results obtained in Fig. 12, we can see that when the node G tries to monopolize the medium for its own traffic (Full-greedy), its throughput gain is more than twice of the one earned in the W-behaved case and the bandwidth gained by its next hop node B is decreased to less than half. Consequently, the end-to-end delay of the flow f_1 is doubled along with the collapse of the packet delivery ratio as depicted in Table II. Hence, as opposed to WLAN the Full-greedy strategy is inadequate in MANETs since it affects the performance of the traffic flow initiated by the greedy node itself.

As alternative strategies, we have implemented Semi-greedy1 and Semi-greedy2 in which the node G constantly sets its CW_m to 31 and 16, its jam-rate to 0.5 and 0.2, respectively. The results show that in the former strategy node G successfully increases its own throughput and the one of its next hop compared to the W-behaved strategy whereas the throughput of all its neighbors decreases to less than half. The drawback of this strategy is the energy necessary to jam 50 % of CTS packets sent by its 2-hops neighbors. Hence, due to the limited energy in MANETs, this strategy is unsuitable for adoption by the node G. For the latter strategy, node B's throughput is increased considerably along with the delivery ratio compared to the Full-greedy strategy; however, the rapidly changing topology of MANETs makes it inefficient since the chosen jam-rate and CW_m may not produce the same results in different network topologies.

Based on our discussion above, the main issues for choosing a suitable greedy strategy in MANETs are the energy constraints and the rapidly changing topology. To circumvent the limitations of the previous strategies regarding these issues, we apply our proposed method where we have implemented two scenarios Rational1 and Rational2. In Rational1, the greedy node G constructs the conflict graph according to the information acquired from HELLO and TC messages (i.e., the best case in terms of the obtained throughput), whereas in Rational2 it assumes the

DOI: 10.1002/sec

	W-behaved	Full-greedy	Semi-greedy1	Semi-greedy2	Rational1	Rational2
		CW_m = 1	CW_m = 31, jam_rate = 0.5	CW_m = 16, jam_rate = 0.2		
End-to-end delay (seconds)	0.679	1.4295	0.554	1.1388	0.859	0.985
Delivery ratio (%)	79.11	15.95	76.59	46.04	50.59	45.93

Table II. End-to-end delay and packet delivery ratio of flow f_1 under various greedy behavior strategies.

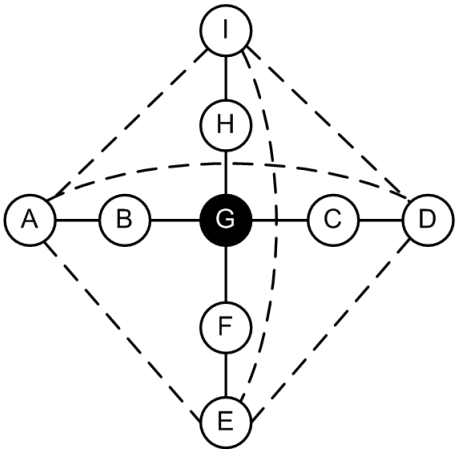

Figure 13. The topology perceived by the node G in the worst case, where the dashed lines denotes the extra links which are not acquired from Hello and TC message

maximum number of contention among the links (i.e., the worst case for the estimated throughput). As shown in Fig. 12 and Table II, both scenarios give good results in terms of end-to-end delay and packet delivery ratio as compared to Full-greedy strategy, with a considerable increase of throughput where node B's throughput is almost equal to the one acquired in W-behaved strategy. Moreover, in both scenarios the greedy node G still gains more bandwidth than its neighbors and maintains a reasonable performance of its flow f_1. Therefore, these results prove the efficiency of our greedy strategy in MANETs.

7.3. Impact of the mobility and network density on the efficiency of our greedy strategy

In order to assess the efficiency of our greedy strategy in dense and highly mobile network, we generate 5 random network topologies consisting of 10, 30, 50, 70 and 100 nodes, respectively. A number of traffic

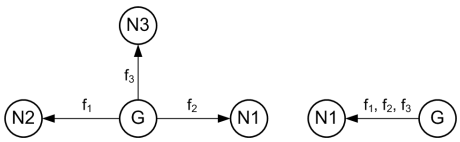

Figure 14. Multiple traffic flows issued from the greedy node G and forwarded either through one or several next hops

flows are also generated in the network such that each source node sends 100 packets per second of 500 bytes each. Some of these flows are generated by the greedy node and forwarded either through one or multiple neighbors.

In these scenarios the nodes move randomly within the area and their velocities vary from 0 m/s to 20 m/s. The network density (D_N) is estimated according to the following formula

$$D_N = \frac{N}{area} \tag{16}$$

where area is the network area in m^2 and N is the number of nodes in the network.

The different values of the network density and the number of traffic flows generated in each scenario are summarized on the Table III.

We define the effectiveness factor (Γ) in order to measure the efficiency of our greedy strategy with the variation of the following metrics: nodes speed, network density and the number of flows. This factor can be expressed as follows;

$$\Gamma = \frac{th_{ng} + \sum_{(i=1)}^{|NH-set|} th_{ni}}{2} - \frac{\sum_{(i=1)}^{N1-(|NH-set|+1)} th_{ni}}{N1-(|NH-set|+1)} \tag{17}$$

Scenario	Network density (D_N)	#flows
1	5.0×10^{-6}	5
2	1.5×10^{-5}	10
3	2.5×10^{-5}	15
4	3.5×10^{-5}	15
5	5.0×10^{-5}	15

Table III. Scenarios setting

such that th_{ng} and th_{ni} are the throughput of the greedy node and its next hop nodes, respectively. $N1$ refers to the number of nodes within the sensing range of the greedy node which are generating traffic flows or forwarding data packets and $|NH - set|$ is the number of next hops of the greedy node.

As we can see from the curves plotted in Fig. 15, the effectiveness factor reduces with the increase of nodes mobility and network density as well as the number of competing flows in the network. Moreover, it is observed that Γ has more decreases in scenarios 4 and 5 where the nodes' speed are 15 m/s and 20 m/s respectively. Despite that our scheme is still effective since the worst value of Γ is 8 packets/s.

As expected, our greedy strategy fails if the greedy node is sending multiple flows simultaneously through several next hops nodes as depicted in Fig. 14, hence even if the greedy node jams the other neighbors' frames its next hops' nodes have to compete with each others to gain access to the medium. Therefore, Γ reduces as the number of flows initiated or forwarded by the greedy node through different next hops's nodes goes to higher. The results graphed in Fig. 16 show that Γ reduces sharply as compared to the results plotted in Fig. 15, in which all the flows initiated from the greedy node are forwarded through one node, because the competition between the next hops nodes leads to a large number of collisions which makes the task of applying our scheme very difficult. Moreover, the increasing velocities of nodes and network density participate also to the failure of our scheme especially in the scenario 5 where the value of Γ is equal to 0 when the velocity of nodes is 20m/s.

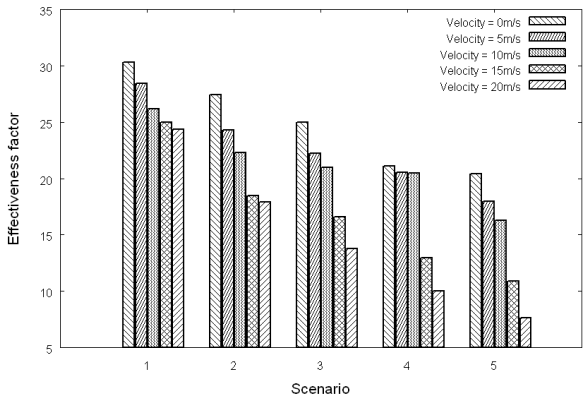

Figure 15. Variation of the effectiveness factor in different scenarios: case of greedy node sending multiple flows through only one next hop node

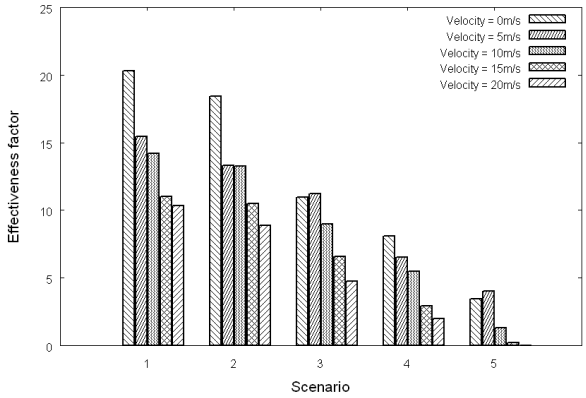

Figure 16. Variation of the effectiveness factor in different scenarios: case of greedy node sending multiple flows through several next hops

8. Conclusion

In this work, we have analyzed the greedy behavior problem in wireless ad hoc networks and proven that its impact can be more devastating compared to that in wireless local area networks. The propagation of the effect of this misbehavior is illustrated through conflict graphs analysis. As a result of this investigation, an effective greedy behavior strategy is proposed suitable for ad hoc networks. This method allows the greedy node to gain more bandwidth share compared to its neighbors and keeps the performance of its ongoing flows reasonable by maintaining its extra bandwidth share within the misbehaving threshold. Our algorithm is evaluated through extensive simulations and the obtained results highlight its advantage in terms of the increase in delivery ratio and the reduction of the end-to-end delays compared to the Full-greedy strategy applied in WLAN.

References

1. A. Wald, "Sequential Analysis", John Wiley & Sons, New York, 1947.
2. F. Massey, "The Kolmogorov Smirnov test for goodness of fit", *ournal of the American Statistical Association*, Vol. 46, No. 253, pp. 68-78, 1951.
3. H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals", *IEEE Transactions on Communications*. Vol. 32, No. 3, march 1984.
4. K. Xu, M. Gerta and S. Bae."Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks", *Ad Hoc Networks Journal*, Vol. 1, No. 1, p 107-123, Jul. 2003.
5. "IEEE 802.11 wireless LAN media access control (MAC) and physical layer (PHY) specifications", ANSI/IEEE Std 802.11, 1999.
6. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 3, March 2000.
7. L. M. Feeney and M. Nilson, "Investigating the energy consumption of wireless network interface in an ad hoc networking environment", *in proc. of IEEE INFOCOM 01*, Anchorage, Alsaska, Apr. 22-26, 2001.
8. Y. Wang, J.J. Garcia-Luna-Aceves, "Performance of Collision Avoidance Protocols in Single-Channel Ad Hoc Networks", *in Proc. of the 10 th IEEE International Conference on Network Protocols (ICNP'02)*, Paris, France , Nov. 12-15, 2002.
9. V. Gupta, S. Krishnamurthy and M. Faloutsos, "Denial of service attacks at the MAC layer in wireless ad hoc networks", *In Proc. of Military Communication Conference MILCOM 02*, Anaheim, CA, Oct. 2002.
10. J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions", *In Proc. of the* 12^{th} *USENIX Security Symposium*, Washington, DC, USA, Aug. 2003.
11. K. Jain, J. Padhye, V. Padmanabhan and L. Qiu," Impact of Interference on Multi-hop Wireless Network Performance", *In Proc. of the* 9^{th} *Annual International Conference on Mobile Computing and Networking (MobiCom 03)*, San Diego, California, USA, Sep. 14-19, 2003.
12. T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", *IETF RFC 3626 (Experimental)*, Oct. 2003.
13. Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworksfor wireless ad-hoc networks", *in Proc. of IEEE INFOCOM 04* , Hong Kong, Mar. 7-11, 2004.
14. A. A. Cardenas, S. Radosavac and J. S. Baras, "Detection and Prevention of MAC layer misbehavior in Ad Hoc Networks", *In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Washington DC, USA, Oct. 25, 2004.
15. M. Cagali, S. Ganeriwal and I. Aad, "On selfish behavior in CSMA/CA networks". *In Proc. of IEEE INFOCOM 2005*, Miami, USA, Mar. 13-17, 2005.
16. P. Kyasanur, and N. H. Vaidya, "Selfish MAC Layer misbehavior in Wireless Networks". *IEEE Transactions on Mobile Computing*, Vol. 4, No. 5, p 502-516, Sept./Oct. 2005.
17. M. Raya, I. Aad, J. P. Hubaux and A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots". *IEEE Transactions on Mobile Computing*, Vol. 5, No. 12, p 1691-1705, Dec. 2006.
18. Y. Rong, S. K. Lee and H. A. Choi, "Detecting Stations Cheating on backoff Rules in 802.11 Networks Using Sequential Analysis".' *In Proc. of IEEE INFOCOM 06*, Barcelona, Spain, Apr. 23-29, 2006.
19. E. Tomita, A. Tanaka and H. Takahashi, "The Worst-Case Time Complexity for Generating All Maximal Cliques", *Theoretical Computer Science Journal of Elsevier*, Vol. 363, No. 1, p 28-42, Oct. 2006.
20. A. L. Toledo and X. Wang, "Robust Detection of Selfish Misbehavior in Wireless Networks". *IEEE Journal of Selected Areas in Communications (JSAC)*, Vol. 25, No. 6, p 1124-1134, Aug. 2007.
21. S. Radosavac and J. S. Baras, "Application of Sequential Detection Schemes for Obtaining Performance Bounds of Greedy Users in the IEEE 802.11 MAC". *IEEE Communications Magazine*, Vol. 46, No. 2, p 148-154, Feb. 2008.
22. L. Guang, C. M. Assi and A. Benslimane, "Enhancing IEEE 802.11 Random Backoff in Selfish Environments", *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 3, p 1806-1822, May 2008.
23. S. Djahel and F. Nat-Abdesselam, "A Fuzzy Logic Based Scheme to Detect Adaptive Cheaters in Wireless LAN". *In Proc. of International Conference on Communications (ICC 09)*, Dresden, Germany, Jun. 14-18, 2009.
24. W.Y. Lee, "Mobile Communications Engineering", McGraw-Hill 1982, ISBN 0-07-037039-7.
25. OPNET Technologies. *OPNET Modeler.* http://www.opnet.com/.

DOI: 10.1002/sec