Article accepted for publication in IEEE Communications Magazine
DOI: 10.1109/MCOM.2014.6829968 © IEEE 2014

# Selfish Insider Attacks in IEEE 802.11s Wireless Mesh Networks

Szymon Szott, AGH University of Science and Technology

**Abstract.** The recently released IEEE 802.11s standard for wireless mesh networks does not provide incentives for stations to cooperate and is particularly vulnerable to selfish insider attacks in which a legitimate network participant hopes to increase its QoS at the expense of others. In this tutorial, we describe various attacks that can be executed against 802.11s networks and also analyze existing attacks and identify new ones. We also discuss possible countermeasures and detection methods and attempt to quantify the threat of the attacks to determine which of the 802.11s vulnerabilities need to be secured with the highest priority.

## I. Introduction

Wireless mesh networks (WMNs) are a potentially important technology for providing Internet access in the near future. This can be attributed to the continuous increase in wireless transmission speeds and the declining cost of devices. The IEEE supports these developments through the constant evolution of the 802.11 standard [1]. One of its latest amendments, 802.11s, is specifically targeted at WMNs and provides the necessary wireless multihop functionality to Wi-Fi devices.

WMNs face multiple security challenges, including susceptibility to selfish (noncooperative) behavior. Selfish attacks are performed by insiders—stations that have already been authenticated and are a legitimate part of the network, with a goal of directly or indirectly increasing their quality of service (QoS) by abusing network mechanisms[1]. This is in contrast to malicious attacks that aim at destabilizing network performance. Malicious attacks have been well studied, whereas selfish attacks are an emerging threat for WMNs for several reasons: equipment vendors may attempt to illegitimately increase the performance of their devices [2]; there is a trend toward ensuring the flexibility of wireless card drivers [3], which paves the way for nonstandard and noncooperative behavior; and although 802.11s provides authentication and encryption to protect the network from external attacks, it is still susceptible to insider attacks, especially of a selfish nature. Community networks (Fig. 1) are especially prone to such attacks because mesh stations can be selfishly configured by their users to raise the QoS (e.g., increase throughput or decrease delay) of all traffic flows terminating in their homes.

[1]In general, other selfish goals may be considered, such as conserving energy. Because mesh stations are usually connected to a mains power supply, we limit the scope of this tutorial to attacks that increase QoS.

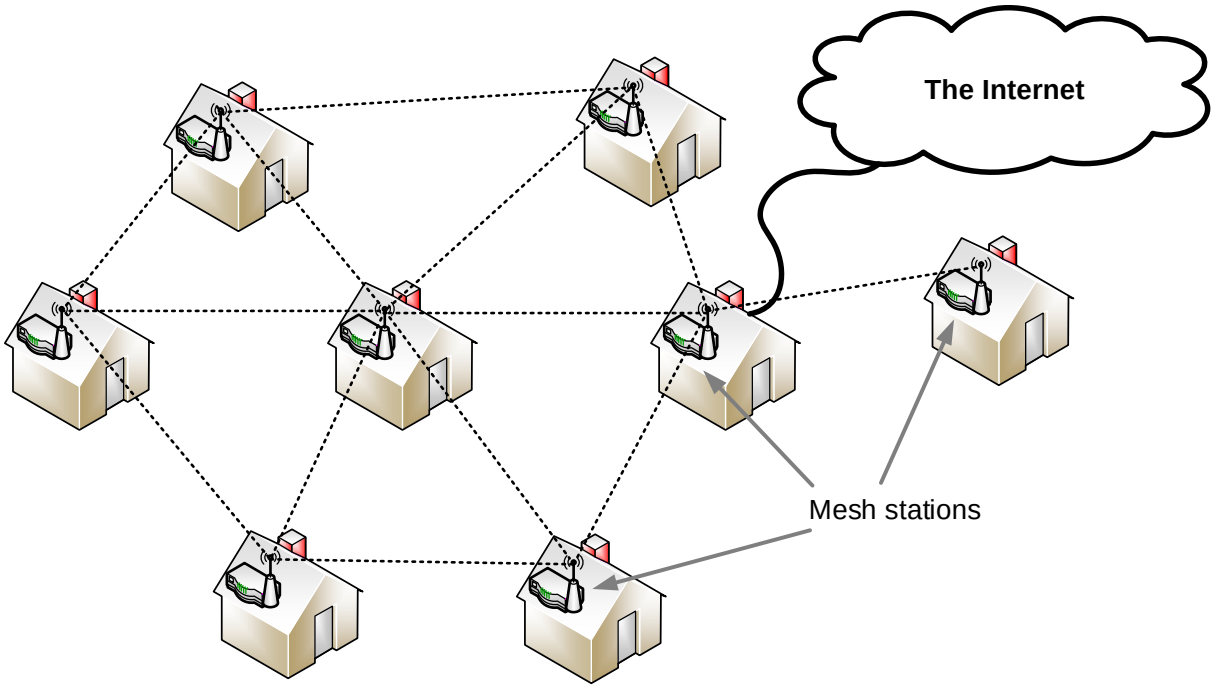


*Fig. 1 Community mesh network—mesh stations placed in each home distribute Internet access.*

The research area of selfish attacks has been well explored for wireless infrastructure and ad hoc network topologies. However, mesh networks, while sharing many similarities with such topologies also possess several distinct characteristics. In principle, they are more structured than ad hoc networks and can support more complex protocols. This is reflected in the 802.11s amendment, which has not yet been analyzed from an insider attack perspective. Existing related security analyses have focused on generic mesh network topologies [4] or considered only malicious insider attacks causing network disruption [5]. This tutorial is the first work which focuses on 802.11s to examine its key characteristics and identify exploits subject to selfish attacks. Additionally, this tutorial aims to foster discussion on insider security in 802.11s and other network types susceptible to selfish attacks.

The contributions of this tutorial are as follows. In Section II, we cover the basic functionality of 802.11s that is required to understand the nature of the selfish attacks. We then discuss attacks against the various mechanisms of 802.11s in Section III. We consider attacks known from the literature, hitherto analyzed for ad hoc networks, and show how they can be executed in 802.11s networks. Furthermore, we identify new attacks inherent to 802.11s. In Section IV, we provide an analysis of countermeasure methods to prevent selfish insider attacks. The main outcome of the paper is the comparative study of the attacks presented in Section V. We conclude the paper with Section VI, where we outline future research directions.

## II. Basics of 802.11s

802.11s describes a set of enhanced functions to provide mesh connectivity for Wi-Fi devices [6]. The whole network consists of mesh stations (Fig. 1) that can be collocated within a single physical device with (a) access points to provide connectivity for nonmesh home Wi-Fi devices

and/or (b) an 802.11 portal to provide Internet connectivity. Throughout the paper, we refer to the latter devices as *gateways*.

The most important mechanisms introduced in 802.11s are related to routing, medium access, peering, authentication, and encryption. The former two are described in detail below. Other mechanisms, such as peering, are briefly addressed in Section III.D. The security mechanisms (for authentication and encryption) protect the WMN from outsider attacks and are therefore out of the scope of this tutorial.

## A. Routing

The hybrid wireless mesh protocol (HWMP) is responsible for routing traffic in 802.11s and is referred to therein as a path selection protocol because it specifies only the PHY and MAC layers. HWMP operates on MAC instead of IP addresses but is not functionally different from a routing protocol. Because HWMP shares similar security threats as (network layer) routing protocols for WMNs, throughout the paper, we use the term routing instead of path selection.

HWMP is based on ad hoc on-demand distance vector (AODV) routing that is extended by adding a tree-based proactive mode and using an airtime metric for determining the shortest route. This metric reflects the amount of channel resources required for transmitting a frame over a given link. In the AODV-based reactive mode of operation, a path for communication between two mesh nodes is set up as required. Path request (PREQ) messages are flooded throughout the network until they reach the intended destination. A path reply (PREP) message is sent back over the best route.

An alternative, proactive mode can be used in scenarios where the network has an Internet gateway (Fig. 1). Because most traffic will flow to and from the gateway, it is useful for each mesh station to have a path setup before communication. Therefore, a routing tree is formed with the gateway as the root. The root periodically disseminates information regarding its availability by either sending PREQs to every mesh station or broadcasting route announcement (RANN) messages, which are then used to initiate a path setup by the stations.

## B. Medium Access

Medium access in 802.11s is governed by the mesh coordination function (MCF), which combines the well-known contention-based scheme of enhanced distributed channel access (EDCA) with a new contention-free scheme called MCF coordinated channel access (MCCA). Because MCCA builds on top of concepts from EDCA, we briefly describe the most important aspects of EDCA and then summarize the operation of MCCA.

### 1) EDCA

EDCA uses four access categories (ACs) to provide traffic differentiation: voice (Vo), video (Vi), best effort (BE), and background (BK). Each category has its own set of medium access parameters: the arbitrary inter-frame space number ($AIFSN$), the contention window minimum and maximum values ($CW_{MIN}$ and $CW_{MAX}$), and the optional transmission opportunity limit ($TXOP_{Limit}$).

In EDCA, before channel access, a station randomly selects a value from the range $[0, CW]$ (initially, $CW = CW_{MIN}$). The chosen backoff value denotes the time slot in which the station will begin its transmission and begins to decrease when the channel has been idle for an $AIFS$

period[2]. The countdown is paused when the channel is sensed as busy. When the backoff value reaches zero, the station starts to transmit and may continuously transmit frames (separated by $SIFS$ periods) within the $TXOP_{Limit}$. If a collision occurs during transmission, $CW$ is doubled until it reaches $CW_{MAX}$. In the case of a successful transmission, $CW$ is reset to the value of $CW_{MIN}$. Otherwise, after a given number of unsuccessful transmission attempts, the frame is dropped.

### 2) MCCA

MCCA provides contention-free transmission using a resource reservation mechanism[3]. Each reservation consists of a set of time intervals, called MCCA transmission opportunities (MCCAOPs), during which the MCCAOP owner (the station that performed the reservation) may transmit to the MCCAOP responders (the stations that receive the reservation request).

Each reservation, set up using dedicated management frames, defines a regular schedule of MCCAOPs. These MCCAOPs are synchronized within a superframe-like period called the delivery traffic indication message (DTIM) interval. At the start of the MCCAOP, the owner temporarily sets its EDCA parameters to $AIFSN = 1$ and $CW = 0$ to ensure high priority in accessing the channel (Fig. 2). Other MCCA stations may not transmit during these reserved periods and may not initiate a transmission that would overlap with such a period (denoted in Fig. 2 as the *silent period*). However, non-MCCA stations, unaware of the reservations, may still interfere by transmitting (thus making MCCA a *soft*-reservation mechanism).

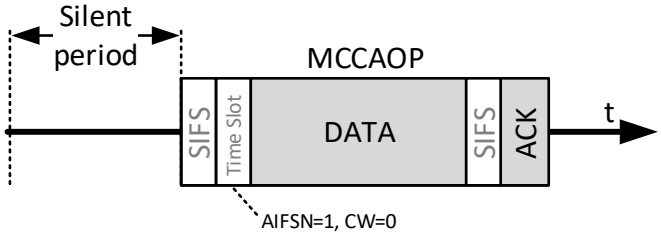


*Fig. 2 Components of an MCCAOP reservation.*

To ensure reservation robustness, all MCCA stations exchange and track information about existing reservations within a two-hop neighborhood. During the reservation setup procedure, each new request is checked against existing reservations so as not to cause collisions. Each station independently calculates its medium access fraction ($MAF$)—the fraction of time reserved within a DTIM by MCCAOPs. New reservations cannot be accepted if, for either the owner or the responders, the $MAF$ would exceed a predefined $MAF_{Limit}$. To this end, each station updates its local $MAF$ as the reservation requests in its neighborhood are set up or torn down and also records the $MAF$ of its neighbors that is disseminated in management frames.

[2] $AIFS$ is calculated as the sum of $SIFS$ and an $AIFSN$ number of time slots (TSs).

[3] An overview of similar reservation-based QoS MAC protocols can be found in [7].

# III. Selfish Attacks

In this section, we cover attacks against routing (Section III.A), medium access (Sections III.B and III.C), and other mechanisms of 802.11s (Section III.D). The overall goal of all described attacks is to increase the attacker's QoS (especially throughput), which can be achieved by directly affecting the local traffic (traffic originating from the client stations connected to the mesh station) and by affecting forwarded traffic to limit the competition for network resources, which has been shown to increase an attacker's effective throughput [8].

## A. Attacks Against HWMP

HWMP is susceptible to many types of routing attacks that have been reported in the literature, mostly malicious attacks that disturb network operation, such as route disruption, route diversion, routing loops, and request floods [9]. However, the main goal of selfish attacks against HWMP is to limit the contention from the forwarded traffic, which can be achieved by rerouting traffic beyond the attacker (and not through the attacker, as in malicious attacks). An attacker may modify PREQs before forwarding them by decreasing their sequence number or increasing the metric to achieve **route diversion** (Fig. 3). An attacker may also perform **route disruption** by dropping management frames, such as RANNs or PREQs, to/from the gateway (Fig. 4), effectively paralyzing path establishment.

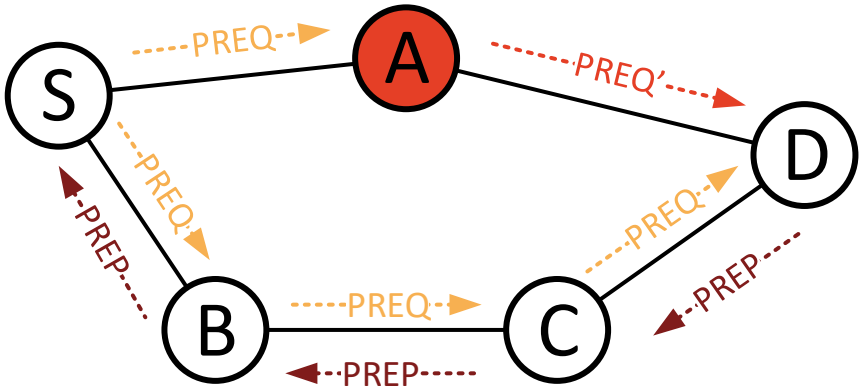


*Fig. 3 Route diversion: station A modifies the forwarded PREQ by decreasing its sequence number or increasing the metric.*

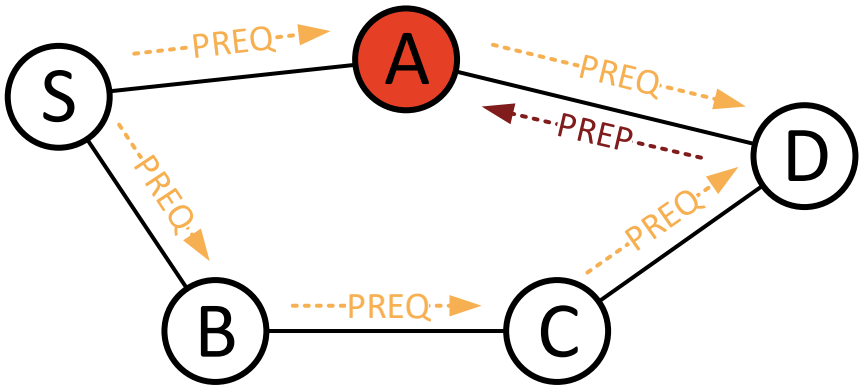


*Fig. 4 Route disruption: station A drops valid PREPs paralyzing path establishment.*

Alternatively, an attacker may perform **selective forwarding** to drop data frames, especially if the gateway is the source or destination. Hitherto, selective forwarding was mostly considered as a malicious attack and studied to determine how it disrupts the network and not how beneficial it is for the attacker [4]. A study showed that it could lead to a *decrease* in the attacker's throughput [10] if the mesh uses a single channel because, owing to the attack, alternative paths are set up through the attacker's neighbors. This increases the contention in the channel and decreases the attacker's effective throughput. However, if multiple channels are used, benefits from this attack can be expected, depending on network topology and traffic patterns. Additionally, a variant of this attack can be targeted at TCP flows, which suffer a decrease in throughput if several consecutive TCP packets are dropped [5].

## B. Attacks Against EDCA

One class of selfish attacks in EDCA constitutes the deliberate **change of medium access parameters**, e.g., shortening $CW$ or $AIFS$ or expanding the $TXOP_{Limit}$ of the AC it is using. This attack increases a station's medium access probability and, hence, may increase its QoS. Such attacks have been well studied in the literature, although mostly for single-hop networks [11]. In mesh topologies, the impact of this attack is limited to the first hop (the selfish attacker being the sender with modified parameters).

The introduction of separate ACs in EDCA paves the way for a second and lesser-known class of attacks called **traffic remapping** [12]. Such attacks consist in modifying the QoS designation of transmitted traffic so that it can be mapped by EDCA onto a higher-priority AC. This attack is similar to the parameter modification in that it also increases medium access probability. Although the parameter space is decreased (there are only four ACs, much less than the possible combinations of $AIFS$, $CW_{MIN,MAX}$, and $TXOP_{Limit}$), this attack is much easier to perform, e.g., using packet mangling software (such as Linux *iptables*), and does not require access to the wireless card driver. In multihop networks, this attack displays further advantage compared with the parameter modification attack; the traffic flow retains the increase in QoS because it uses a higher AC. Additionally, in multihop networks, this attack can be extended to include the downgrading of forwarded traffic priority [8].

## C. Attacks Against MCCA

The most obvious selfish behavior in any resource reservation protocol is to reserve the maximum amount of resources possible. To a certain degree, this fairness issue is addressed in 802.11s by the $MAF$ mechanism. Studies have shown that this approach may require further refinement [13]. We leave this fairness issue open and concentrate on attacks that are more elaborate. We have identified two classes of selfish attacks that can be performed against the MCCA function: the former allows directly increasing the attacker's QoS, whereas the latter aims to eliminate competition in medium access.

### 1) Disregarding Reservations

An attacker may deliberately **contend during the MCCAOPs of other stations**. This attack is especially profitable in an all-MCCA environment, where the attacker is sure that its transmission will not collide with transmissions from stations unaware of the setup reservations. This attack can have several variants depending on when the attacker begins its transmission with respect to the beginning of the reserved period. We describe them below in their sequential order.

First, the attacker may transmit before the start of the MCCAOP (in the silent period, Fig. 2). Since MCCA-enabled stations cannot initiate a transmission if it would interfere with an MCCAOP, this period is free from contention. Thus, the attacker is guaranteed a high probability of successful transmission. Such an attack can be executed either blindly, by simply ignoring reservations, or expertly, by timing the transmissions to always begin just before the beginning of the MCCAOP. Because it is necessary for the channel to be idle for at least a $SIFS$ period before the transmission, in the extreme case, the attacker may set $AIFSN = 0$ and $CW = 0$ at the start of the MCCAOP, thus preempting the MCCAOP owner's transmission (which sets $AIFSN = 1$).

Second, the attacker may transmit in parallel to the MCCAOP owner by setting the exact same parameters ($AIFSN = 1$ and $CW = 0$). This attack has a high risk of collision because the owner will likely use its reserved period. However, it can be a good method for jamming streams, ultimately leading to their teardown (Section III.C.2).

Finally, the MCCAOP owner may have nothing to transmit. During such empty MCCAOPs, the owner may send a QoS Null frame to end the MCCAOP or refrain from sending such a frame when it is not needed or undesirable. Therefore, in the latter case, an attacker may opportunistically transmit its data. Such an attack can be implemented by setting $AIFSN > 1$ and $CW = 0$ at the start of the MCCAOP and could potentially increase the network performance.

### 2) Limiting Competing Reservations

There are several ways in MCCA to eliminate contending reservations. First, an attacker may both **disregard its $MAF$ during internal checks and disseminate a high $MAF$** value to its neighbors. Such an attack eliminates neighbors from reserving the attacker's resources (which means less forwarded traffic) and allows the attacker to request as many reservations as possible, given its neighbors' $MAF$ values.

Another method to block neighboring stations is increasing their $MAF$ value by **disseminating false reservation advertisements** that can report reservation periods, of which the attacker is an owner or responder, and interference caused by reservations in the attacker's neighborhood. The latter case enables the attacker to use nonexisting station identifiers to legitimize the false advertisement.

The two methods described previously are an indirect way of limiting contending reservations. To directly eliminate contention from neighboring MCCA stations and make room for new reservations, an attacker may attempt to **tear down existing reservations**. The standard states the following reasons for a teardown to occur:

- explicit, through appropriate management frames;
- implicit, through lack of messages received within a certain timeout; and
- from reservation conflicts (caused by overlapping MCCAOPs).

To exploit each of these teardown conditions, an attacker can forge management frames, jam frames (Section III.C.1), or send false advertisements, respectively. Once the reservation has been torn down, the attacker will need to immediately setup its own reservation to make this attack beneficial.

### D. Attacks Against Other 802.11s Mechanisms

802.11s introduces other mechanisms required for the proper operation of a mesh network, such as power management, channel switching, and interworking with external networks. It seems that these mechanisms do not significantly increase the selfish attack space in comparison with the vulnerabilities of routing and medium access. However, there are two more mechanisms that enable selfish attacks as described below.

The mesh peering management protocol allows mesh stations to establish connections with neighbors by performing a handshake using management messages. A selfish attacker, having established a path to the gateway, may perform a variation of the selective forwarding attack and **close the peering with neighbors** who are further away from the gateway. This may eliminate contention from them (especially in multichannel networks, cf. Section III.A). However, this also reduces path redundancy that might cripple the attacker in case of link failures.

802.11s introduces a framework for congestion control to limit the effects of congestion in a mesh network. While monitoring and detecting congestion are not within the standard scope, 802.11s briefly describes the notification mechanism and suggests possible reaction methods. Having detected a particular congestion, a mesh station notifies others, including its neighbors and the traffic source. The message contains the congested destination and the expected duration of the congestion. As a method of reaction, 802.11s suggests discontinuing or reducing the forwarding of traffic to the congested station. The standard provides a vague description of this framework, and if it will be implemented and used is difficult to determine. After all, congestion mechanisms may be implemented in higher layer protocols. The details of a selfish attack against these mechanisms would be implementation specific; however, in principle, it may allow the attacker to **disseminate false congestion information** to silence other mesh stations.

## IV. Preventing Selfish Attacks

Various countermeasures can be used to prevent selfish attacks in wireless mesh networks. Fig. 5 provides a classification of such methods with a brief description, requirements, and examples for each. The three main approaches are to *prohibit* an attack, *mitigate* the attack's impact during its span, or *incentivize* stations to refrain from attacking. The presented classification is generic enough to be applicable to any network type threatened by selfish attacks.

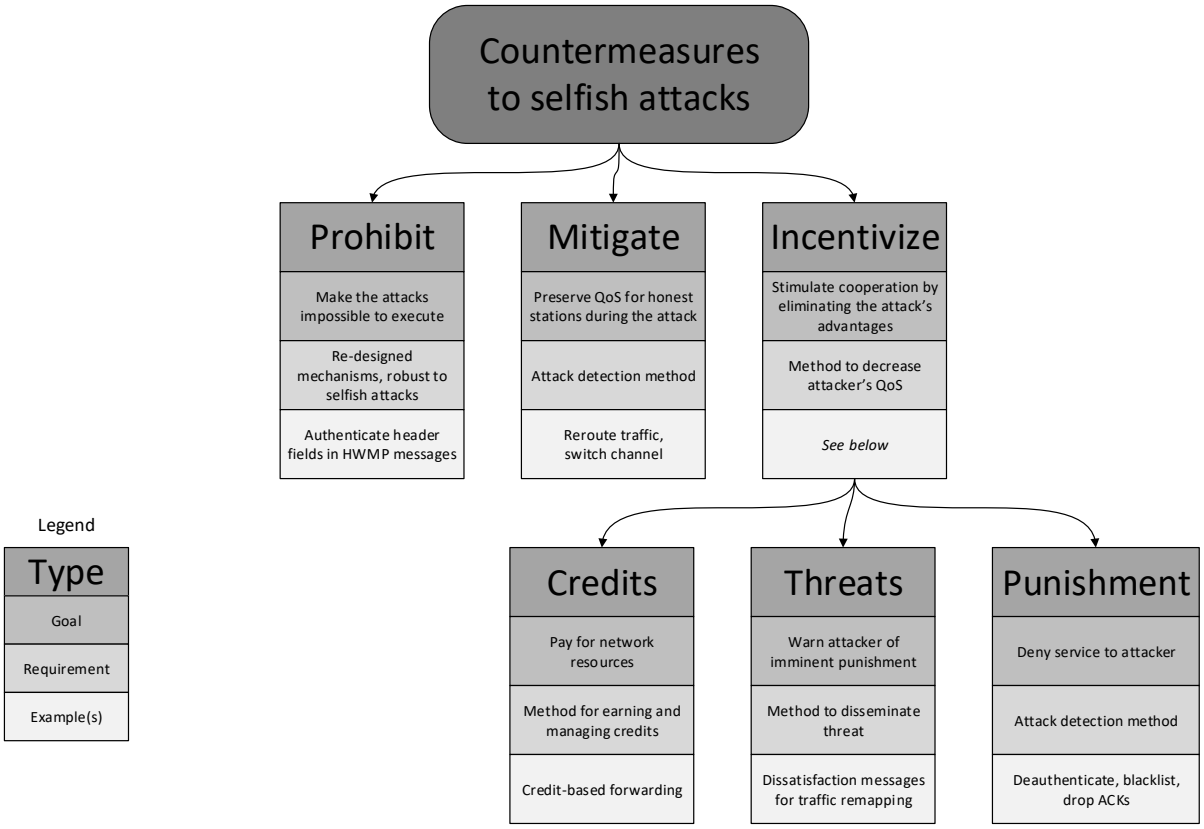


*Fig. 5 Classification of countermeasures to selfish attacks in 802.11s.*

Ideally, selfish attacks should be prohibited by constructing robust mechanisms where the attack is either made impossible by design or has no effect and can thus be ignored. This can be achieved by resorting to tamper-proof hardware, within which some functionality (such as the medium access function) is encoded. This approach is not practical as it increases the cost and leads to a loss of flexibility, which is a desired and emerging trait of Wi-Fi networks [3]. Alternatively, cryptographic measures can be used. Secure HWMP [9] prevents unauthorized manipulation of routing messages through additional authentication and encryption and prohibits many malicious routing attacks, the main goal of the protocol, and two of the three identified selfish attacks against HWMP: route diversion and route disruption. However, whenever the mechanisms defined in 802.11s are redesigned to prohibit selfish attacks, this breaks compatibility with existing devices. Therefore, prohibiting selfish attacks is not always a practical approach.

If selfish attacks cannot be prohibited, it is useful to mitigate their impact so that the QoS of nonattacking stations is preserved. This can be considered a specific type of fault tolerance. In most cases, this means isolating the attacker, e.g., by rerouting traffic or using a different channel on a link in the attacker's neighborhood. Mitigation requires a method for detecting the presence of an attacker. This can be either direct or indirect (through QoS loss of the attacker's neighbors). We discuss detection issues later in this section.

Selfish attacks can also be prevented when the attacker knows, or can learn, that they are not beneficial. This is the premise of incentive-based mechanisms, which coax mesh stations into cooperation by relying on the operant conditioning of user behavior. Such mechanisms can be based on credits, threats, or punishment and are often the most practical solution for WMNs.

In credit-based incentive schemes, stations receive rewards (credits) for participating in the operation of the network, e.g., for forwarding traffic. Credits can be spent when using network resources (e.g., to pay for forwarding local traffic). This is a potentially valid method for community mesh networks where each node generates local traffic and relays the traffic of neighbors. However, such methods are subject to the station's placement in the network topology. An outlying station may have no relay traffic, whereas a well-placed attacker may have enough traffic to forward to afford time-limited attacks.

The threat of punishment can also be used to incentivize selfish attackers, which requires a method for disseminating the information that the punishment is imminent, e.g., by stations that either have detected the attack or are suffering because of it. This approach has been shown to be a good incentive in the case of traffic remapping attacks [12]. Detecting such attacks is cumbersome (it incurs a large overhead), but if the attack causes no harm, then it can be allowed. In the proposed approach [12], additional dissatisfaction messages signify that a station is suffering because of this attack and that there is an imminent threat of punishment. Such messages can be ignored by honest stations and allow the attack to persist if they are not present. Studies show that the proposed approach leads to operating points where the attack is either counterproductive or harmless.

In the third incentive-based method, the attacker, having been identified, is punished by using some form of denial of service. These include tit-for-tat behavior (e.g., also dropping frames), banning the attacker from the network (by deauthentication and blacklisting), or using existing mechanisms (e.g., congestion control) to limit the effectiveness of the attack.

Several of the preventive methods described above require the identification of the attacker before they may be applied. Detection methods fall into three categories: passive (does not require additional functionality), active (requires the implementation of extra mechanisms), and hybrid (combining both approaches).

The passive detection of selfish attacks is referred to as the watchdog method [5], which is based on promiscuous channel monitoring to determine the behavior of neighboring stations and works best for MAC-layer attacks and also for attacks against the routing mechanism. The accuracy of the method depends on the attack type and the network configuration. For example, it may become problematic to perform accurate observations in a multichannel network. The watchdog mechanism can be coupled with traffic classification mechanisms to detect traffic remapping attacks.

Alternatively, new mechanisms may be introduced to existing protocols to provide active detection of specific attacks. For example, additional end-to-end acknowledgment schemes can facilitate the detection of the selective forwarding and route disruption attacks [14].

Hybrid detection schemes are usually a combination of the watchdog mechanism described previously and a reputation scheme [14]. Stations calculate a reputation score of other mesh stations based on first-hand (using the watchdog approach) and optional second-hand information, disseminated by their neighbors. Stations with a low reputation score are identified as attackers.

# V. Comparative Analysis of Attacks

With the selfish attacks as basis, we use an attack tree (Fig. 6), which is a method of security analysis described by Schneier in [15]. The attacks are identified by their common goal (increasing the attacker's QoS) and divided into direct (affecting local traffic) and indirect (affecting forwarded traffic) attacks. The attacker can choose one of the 802.11s mechanisms (EDCA, MCCA, HWMP, peering, and congestion control). Each attack is subjectively rated by its execution cost $c$, risk of being detected $r$, potential QoS gain $g$, and the aggregated threat $t = \frac{g}{cr}$. The cost, risk, and gain are assessed on a scale of 1 to 4, with 1 being very low and 4 being very high. Additionally, we have numbered the attacks for easier identification.

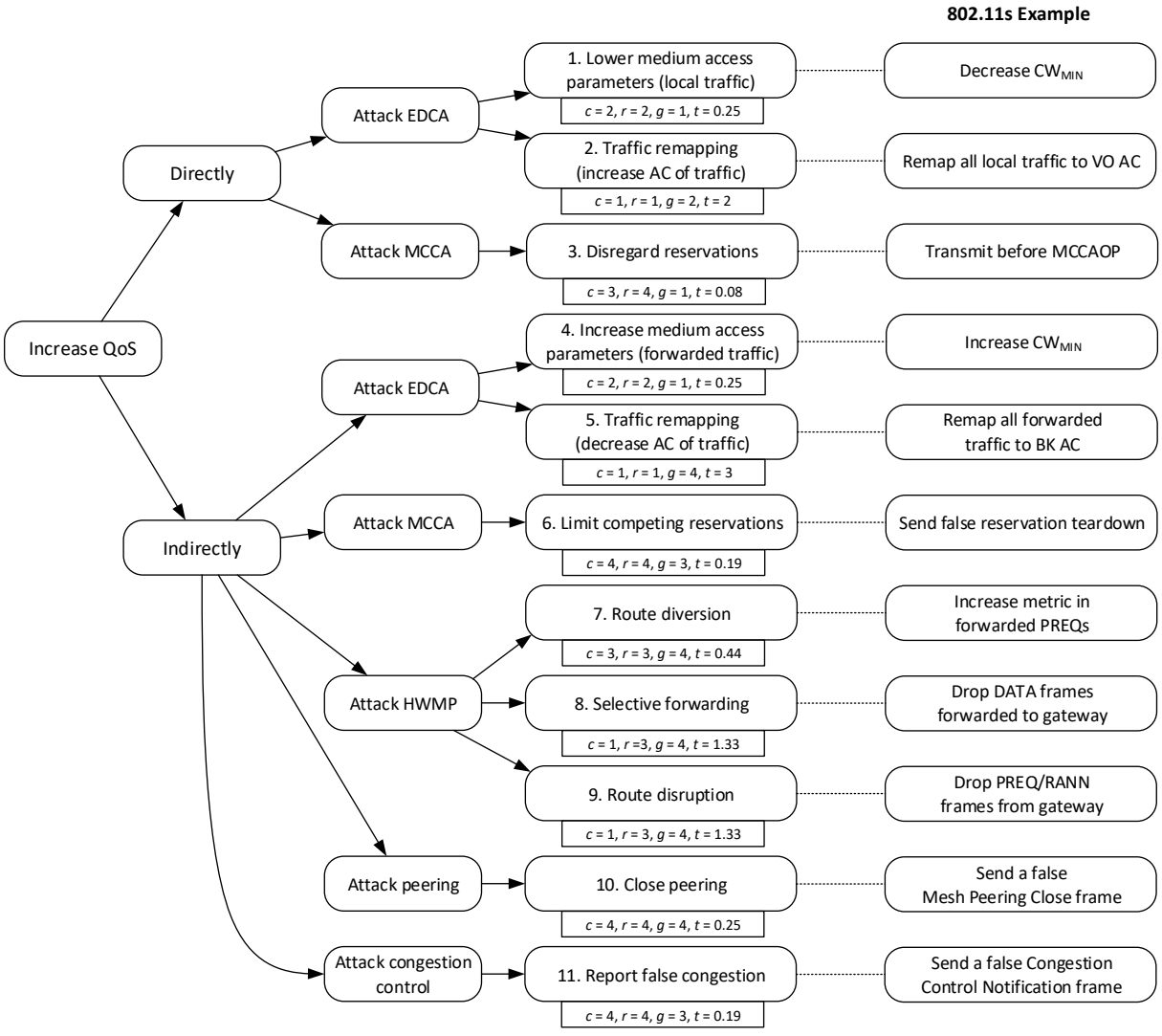


*Fig. 6 The attack tree for selfish attacks in 802.11s networks.*

The execution cost $c$ is related to the attack's technical requirements. Attacks 2, 8, and 9 ($c = 1$) require only packet mangling software (such as Linux *iptables*), which may be independent of the wireless card driver. Attacks 1 and 4 ($c = 2$) require drivers that allow the configuration of selected parameters. Attacks with an even higher cost require fully flexible drivers [3] either to change the attacker's internal behavior (attacks 3 and 7 with $c = 3$) or to successfully perform packet injection in the hope of changing the behavior of others (attacks 6, 10, and 11 with $c = 4$).

The detection risk of an attack $r$ depends on the required discovery method. Attacks 3, 6, 10, and 11 carry the highest risk ($r = 4$) because a simple watchdog mechanism can detect message forgery or any other direct violation of the standard. Attacks 7 through 9 ($r = 3$) introduce only local and subtle changes and require a more detailed watchdog or an active detection mechanism (such as the end-to-end acknowledgements previously described). Attacks 1 and 4 ($r = 2$) require not only a watchdog but also statistical data owing to the random nature of CW monitoring. Finally, attacks 2 and 5 have the lowest risk ($r = 1$) because they require traffic classification that consumes more time to be accurately detected.

The gain $g$ is assessed in terms of the increase in QoS for the attacker. Gains are highly dependent on the network scenario, and more research is required to quantify them. However, we have attempted to group the attacks using the following reasoning. First, we notice that direct attacks can only impact the uplink QoS parameters, whereas indirect attacks can also impact the downlink QoS parameters [8]. Because, in community networks, users are more likely to prefer having better downlink QoS, we rate indirect attacks higher. Among the indirect attacks are those that are able to directly reduce the amount of forwarded traffic (5 and 7 through 10, $g = 4$). The remaining indirect attacks (6 and 11) attempt to influence other stations to decrease the amount of generated traffic ($g = 3$). Next, the single direct attack with a multihop impact (no. 2) has $g = 2$. Direct attacks with only a single-hop impact (1, 3, and 4) have the lowest gain.

In terms of threat, the two traffic remapping attacks (2 and 5) clearly have the highest score owing to their combination of low cost, low risk, and high gain. This new type of attack is followed by well-known routing attacks related to packet dropping. On the other hand, attacks 3, 6, and 11 constitute a low threat and can be considered unlikely.

# VI. Conclusions

Based on the performed analysis, we conclude that 802.11s networks are indeed susceptible to selfish insider attacks. These attacks constitute new threats either because they exploit new vulnerabilities or because well-known attacks are executed in a new, multihop context. Using the *attack tree* method we have quantified the threat of the attacks and showed which of them should be addressed. This tree can be useful for any mesh network where it can be adapted by setting appropriate weights.[4] Further research is required to determine the exact impact of the attacks. Analytical models, simulation studies, and experiments performed using real-world 802.11s equipment should provide additional insights.

# Acknowledgements

This work was carried out as part of a project financed by the Polish National Science Centre (decision no. DEC-2011/01/D/ST7/05166).

[4]Indeed, the attack tree is relevant to any network type that is susceptible to selfish insider attacks. For example, selective forwarding attacks can occur in any multihop wireless network.

# Bibliography

[1] “IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, March 2012.” 2012.

[2] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, “Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards,” in *Proc. of INFOCOM*, 2007, pp. 1181–1189.

[3] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, “Wireless MAC processors: Programming MAC protocols on commodity Hardware,” in *INFOCOM, 2012 Proceedings IEEE*, 2012.

[4] N. Ben Salem and J.-P. Hubaux, “Securing wireless mesh networks,” *Wireless Communications, IEEE*, vol. 13, pp. 50–55, 2006.

[5] L. Lazos and M. Krunz, “Selective jamming/dropping insider attacks in wireless mesh networks,” *Network, IEEE*, vol. 25, no. 1, pp. 30–34, 2011.

[6] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, “IEEE 802.11s: The WLAN Mesh Standard,” *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 104–111, 2010.

[7] M. Natkaniec, K. Kosek-Szott, S. Szott, and G. Bianchi, “A Survey of Medium Access Mechanisms for Providing QoS in Ad-Hoc Networks,” *Communications Surveys Tutorials, IEEE*, vol. 15, no. 2, pp. 592–620, 2013.

[8] S. Szott, M. Natkaniec, and A. Banchs, “Impact of Misbehaviour on QoS in Wireless Mesh Networks,” in *Proc. of IFIP Networking*, 2009.

[9] M. Islam, M. Hamid, and C. Hong, “SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks,” in *Transactions on Computational Science VI*, vol. 5730, M. Gavrilova and C. Tan, Eds. Springer Berlin Heidelberg, 2009, pp. 95–114.

[10] K. Gierlowski and J. Konorski, “Router selfishness in community wireless mesh networks: Cross-layer benefits and harms,” in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, 2010.

[11] S. Szott, M. Natkaniec, and A. R. Pach, “An IEEE 802.11 EDCA Model with Support for Analysing Networks with Misbehaving Nodes,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 13, 2010.

[12] J. Konorski and S. Szott, “EDCA remapping in ad hoc IEEE 802.11 WLANs: An incentive compatible discouragement scheme,” in *Wireless Days (WD), 2012 IFIP*, 2012.

[13] S. Chakraborty and S. Nandi, “IEEE 802.11s Mesh Backbone for Vehicular Communication: Fairness and Throughput,” *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 5, pp. 2193–2203, 2013.

[14] J. Konorski, “Effective Data-Centric Reputation Systems for MANETs: A Novel Evaluation Framework,” in *Communications (ICC), 2011 IEEE International Conference on*, 2011, pp. 1–6.

[15] B. Schneier, “Attack trees,” *Dr. Dobb’s journal*, vol. 24, no. 12, pp. 21–29, 1999.