Krichene and Boudriga *EURASIP Journal on Wireless Communications and Networking* 2013, **2013**:100
http://jwcn.eurasipjournals.com/content/2013/1/100

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Intrusion tolerant QoS provision in mobile multihop relay networks

Neila Krichene* and Noureddine Boudriga

Abstract

We propose in this article an architecture called trusted timely attacks-tolerating communication architecture (3TCA) designed for mobile multihop relay (MMR) networks. The 3TCA architecture is based on trusted components providing trusted time-related and security-related services along with an intrusion-related service while guaranteeing QoS compensation. 3TCA components are implemented at access nodes such as relay stations (RSs) and multihop relay base stations (MR-BSs) in order to achieve trusted QoS-aware handover management while guaranteeing intrusion tolerance. In particular, we provide QoS guarantees in terms of delay, bandwidth and jitter for the MMR networks while addressing group mobility through performing handover disturbance compensation for mobile RSs. Meanwhile, we tolerate particular timely-based and DoS attacks through compensating their impact on the already agreed QoS level. Simulations show that adopting compensation is suited for the MMR context although it induces additional complexity.

Keywords: QoS, Intrusion tolerance, Multihop relay networks, 3TCA architecture

1 Introduction

The last years have been marked by a growing need for ubiquitous access to multimedia and real time applications at the metropolitan scale. Mobile subscribers are becoming very particular about the quality of the provided services and the resulting cost. The wireless technology which will succeed in providing the best QoS at the minimum cost while being interoperable with existing technologies will probably gain popularity. The traditional point to multi-point access is not able to achieve high throughput when the mobile subscriber moves away from the base station. Adopting the mesh technology to increase the coverage has its drawbacks since it induces complexity in routing, network management, QoS provision and security guarantees. Recently, the mobile multihop relay (MMR) networks emerged as an optional deployment that can be adopted to enhance the coverage and the performance in an access network. MMR networks specification is detailed within the IEEE 802.16j-2009,[1], which is the new amendment to IEEE 802.16-2009, [2]. A MMR network encompasses a multihop relay base station (MR-BS), one or more fixed or mobile relay stations (rSs) managed by the network operator and subscriber stations (SSs). The RS relays traffic and signaling between the SS and the MR-BS when needed and in more than two hops system, traffic and signaling between an access RS and MR-BS may be relayed through intermediate RSs. Multihop relay networks can be seen as a specialization of mesh networks that intends to reduce the complexity of the mesh mode while enhancing the wireless coverage and performance.

Providing QoS is a crucial issue for supporting advanced Internet applications and satisfying customers requirements. The IEEE 802.16j-2009 amendment, [1], has been designed with QoS in mind. Therefore, five scheduling services have been proposed to address different QoS requirements related to both real time and non real time applications. Each scheduling service provides a specific QoS level to its flows through the definition of specific QoS metrics including rate, latency, and jitter. Nevertheless, the IEEE 802.16j-2009 amendment does not specify any QoS architectures or scheduling mechanisms that implement the proposed services, [1]. Besides, the QoS provision in the multihop relaying context is a complex issue as multiple hops increase delays resulting by adding the processing delay at each node. Meanwhile, nodes and RSs' mobility causes problems related to end-to-end QoS

*Correspondence: neila.krichen@gmail.com
CN&S Research Laboratory, University of Carthage, Carthage, Tunis, Tunisia

provision since the handover operation induces additional delays and requires considerable available resources at the visited access nodes in order to provide the demanded throughput. Last but not least, the intrusion tolerance property may be considered as an additional QoS criterion that needs to be adopted in order to guarantee the provision of secure services.

Meanwhile, the MMR networks should be secure in order to gain the trust of the customers. The intrusion tolerance's aim is to guarantee the security through the provision of unremitting secure services. More precisely, an intrusion tolerant system is a system which continues delivering its services despite the intrusions affecting it as defined by the important work presented in [3]. Since vulnerabilities are always present in each system and since the protection measures countering the intrusions exploiting such vulnerabilities are always imperfect, the intrusion tolerance property guarantees the continuous provision of secure and correct services despite partially successful attacks. The intrusion tolerance property has been implemented in ad hoc and sensor networks but we think that it is important to implement this paradigm in MMR networks. First because we are dealing with a wireless network which faces multiple attacks conducted on the air interface. Second because the MMR networks may be targeted by serious attacks such as DoS ones which may dramatically affect multiple customers at once.

A few research works addressed separately security and QoS optimization within multihop relay networks. For instance, Chang et al. [4] proposed a self-optimization handover mechanism that uses the global positioning system (GPS) navigation system in order to reduce the number of possible handover and optimize the channel scanning procedure. Chang et al. [5] propose an algorithm that minimizes collusion in contention-based initial ranging and bandwidth request in order to achieve fast handover along with low dropping and low collision probability. Kim et al. [6] designed a speed-sensitive handover under hierarchical cellular system that dynamically adjusts the cell size of each cellular layer depending on the distributions of the mean speeds of mobile users in order to increase the channel utilization and optimize the dropping probability of new and handover calls. Last but not least, Ann et al. [7] proposed to secure the routing for IEEE 802.16j networks but they did not consider the provision of QoS.

In this article, we intend to address QoS engineering for MMR networks while taking intrusion tolerance into account. A few intrusion tolerant routing protocols for the ad hoc and mesh contexts have addressed QoS issues, [8,9]. However, such protocols can not be directly adopted by relay networks. For instance, the QoS and intrusion tolerant ad hoc routing (QITAR) protocol that was proposed in [8] is a QoS aware intrusion tolerant routing protocol for ad hoc networks. Nevertheless, QITAR can not be applied in the MMR context as it is very vulnerable to high-speed mobility and it may experience scalability problems. In order to address the mesh context, we proposed an Intrusion Tolerant routing protocol called mesh routing with QoS and intrusion tolerance (MERQIT) that guarantees consistent delay constraints while securing the route establishment procedure and achieving good performance at the metropolitan scale, [9]. MERQIT adopts the concept of clusterheads which are in charge of managing the non line of sight mobile nodes. Although a clusterhead relays traffic; it is an independent mobile node which is not managed by the network operator.

To the best of authors' knowledge, no research work tried to combine QoS provision and intrusion tolerance for MMR networks within the same framework since both properties are considered as conflicting. Meanwhile, there is no research effort in compensating the disturbances induced by the handover procedure and the possible attacks on the already agreed QoS. Besides, the intrusion tolerance concept does not have been applied to the multihop relay context yet. We intend in this article to combine the provision of QoS and the guarantee of the intrusion tolerance property for MMR networks. The first challenge facing us is the secure estimation of the QoS and its guarantee despite the RSs' mobility. Second, we should take up the challenge of guaranteeing the continuous provision of the agreed QoS level despite the success of particular attacks targeting the MMR networks. In order to address these issues, we propose in this article a QoS architecture for MMR networks called trusted timely attacks-tolerating communication architecture (3TCA) which tolerates particular attacks while adopting novel design principles that enhance the scalability and the efficiency of our proposed protocols. We also address the RSs' mobility and intrusion tolerance within multihop relay networks in a mature way as we compensate handover and malicious behavior disturbances on the agreed QoS. Our contributions in this article are the following: first, we propose a trusted and efficient way to estimate the QoS delivered on the wireless edge and core links within the multihop relay network. Second, we provide an optimized mobility management that evaluates the handover disturbance in terms of delay. After that, the disturbance is compensated by accelerating the affected flows on chosen links along the path. Our proposed scheme may also achieve jitter and bandwidth compensation. If a complete compensation is not possible, we propose to come close to the agreed level through re-performing the admission control procedure with updated QoS values. Third, we propose a scheme that detects particular delay-sensitive and DoS attacks and tolerates them through compensating their side effects on the agreed QoS level.

The remainder of this article is organized as follows. First, we present the state of the art by over-viewing the IEEE 802.16j specifications regarding QoS, handover and path management. We also introduce the intrusion tolerance concept while examining some research work, as it is applied to wireless networking and we overview the requirements for an efficient QoS provision. Second, we define the novel 3TCA architecture and we present its services and the QoS compensation it achieves. Third, we detail the 3TCA implementation in Relay Networks. After that, we overview the properties and features of 3TCA-based relay networks. Finally, we evaluate the performance of our proposition and we conclude.

2 State of the art

This section presents the state of the art regarding multihop relay networks along with the appliance of the intrusion tolerance concept within wireless networks and the requirements for an efficient QoS provision.

2.1 IEEE 802.16j specifications regarding QoS, handover and path management

An MMR network encompasses fixed multihop relay base stations (MR-BSs), fixed or mobile relay stations (RSs) managed by the network operator and mobile subscriber stations (SSs) moving at vehicular speed. The wireless link between the SS and the managing MR-BS or RS and the wireless link between the MR-BS or RS and a subordinate RS during network entry are both called "access links" while the wireless link between a MR-BS and a RS or between two RSs is called "relay link" as depicted by Figure 1. Access and relay links can be either uplinks or downlinks. The IEEE 802.16j amendments specify that a dynamic service addition request (DSA-REQ) signalling message is used for admission control and path management in the context of multihop relays with scheduling RSs, [1]. The DSA-REQ can only be sent over relay links from the MR-BS or the RS to its subordinate RS. Generally speaking, a MR-BS sends a DSA-REQ to all RSs on the path to request an admission control decision. That request is processed by each RS on the path and forwarded to its subordinate RS. The DSA-REQ encompasses the service flow parameters described by Type/Length/Value (TLVs) fields. A RS sends a DSA-RSP message to the MR-BS when it can not accept the service flow indicated in the DSA-REQ or in order to confirm the path management operation requested in the correspondent DSA-REQ. On the other hand, the MR-BS should send a DSA-ACK to all RSs on the path upon receiving a DSA-RSP from an access RS for the purpose of admission control. Each intermediate RS processes the DSA-ACK and forwards it to its subordinate RS. The IEEE 802.16j amendments also define the dynamic service change request (DSC-REQ) signalling message which is used for admission control and path management whenever a dynamic change of the parameters of an existing service flow is required.

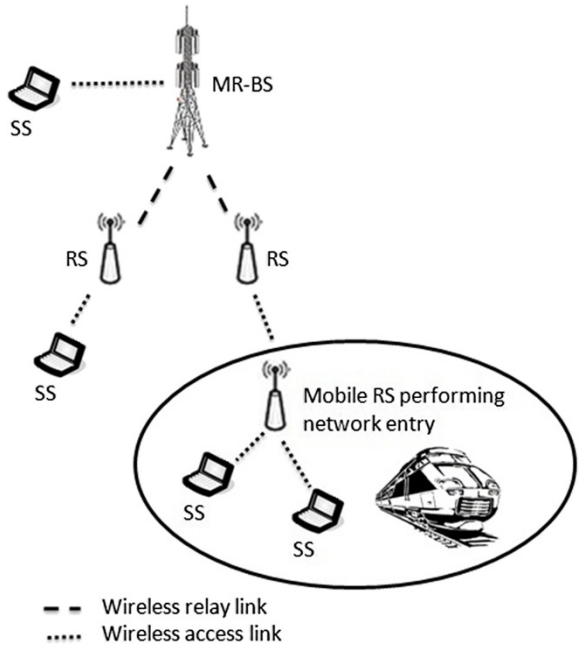

Figure 1 MMR network.

Routing in wireless multihop relay networks is tree-based. In order to address path establishment, maintenance and release, the IEEE 802.16j amendments propose to base routing decisions on metrics such as radio resource availability, radio link quality and traffic load at the RSs and propose to take these decisions at the MR-BS level based on information provided by the RSs, [1]. However, the amendments do not specify how the decisions should be made, [10]. Besides, the standard proposes two approaches to path management namely the embedded path management approach and the explicit path management approach.

The embedded path management approach allocates the connection identifiers (CIDs) in a hierarchical manner. More precisely, the MR-BS allocates CIDs to its subordinate stations so that the CIDs allocated to all subordinate RSs of any station are a subset of the allocated CIDs for that station. Consequently, the path management is simplified because there is no need to store specific routing tables at the RS level and there is a reduced need for signaling to update the path information. On the other hand, the explicit path management approach uses an end-to-end signaling mechanism in order to distribute the routing table along the path. In more details, each path is identified by a path ID to which the CIDs are bound. The MR-BS needs to send the required information to the RSs belonging to the concerned path whenever it is created, updated or removed. Optionally, the MR-BS may specify

the QoS requirements associated with each CID in order to enable the RSs to take the scheduling decision independently in case of distributed scheduling mode. The explicit path management approach needs small routing tables at the RSs and enables a reduction of the overhead required to update these tables, [10].

Authors of [7] propose a path selection method for the non-transparent mode of IEEE 802.16j networks that finds the lowest latency and the best path with high throughput. The proposed method uses metrics such as link available bandwidth, signal-to-noise ratio (SNR) and hop count in order to determine the path cost. The desirable path should then balance the load with other paths fairly by favoring smaller hops and less robust channel coding schemes. Authors of [7] suggest an example network model for applying the proposed method and prove its effectiveness. Nevertheless, the proposed method does not consider the security and may suffer from numerous attacks such as DoS.

Regarding handover, when a mobile RS hands over, all the SSs attached to it should also hand over with it. Besides, the RS-BS can exchange context info on the backbone in order to accelerate the handover process.

2.2 Intrusion tolerance in wireless networking

The fault tolerance concept was firstly introduced in the field of software engineering, [3,11]. Its aim was to preserve the delivery of correct service despite the presence of active faults, [3]. Fault tolerance is fulfilled through the error detection and then the subsequent system recovery. The malicious and accidental fault tolerance for internet applications (MAFTIA) project involved experts from five countries and six organizations and lasted from January 2000 to February 2003 in order to investigate the tolerance paradigm for security, [12]. In more details, the MAFTIA project proposed an integrated distributed architecture based on intrusion tolerant systems and designed the required mechanisms and protocols to form the building blocks of large scale dependable applications. The MAFTIA project main results were the provision of a dependable middelware, dependable trusted third parties, distributed authorization mechanisms and large scale Intrusion Detection Systems. Note that the trusted timely computing base (TTCB) components were part of the proposed architecture.

Most of the research works that are interested in providing intrusion tolerance for wireless networks are targeting wireless sensor networks. For instance, Ma et al. [13] propose a fault-intrusion tolerant framework called WSN-TUM that incorporates both conventional non-security fault and intrusion fault management in order to ensure fault-intrusion tolerant routing. In more details, the WSN-TUM identifies four types of faults that can target WSNs which are hardware faults, software faults, middelware faults, and intrusion faults and then adopts multi-path, multi-version and fragmentation techniques in order to fulfill the intrusion tolerance property. Ma et al. [13] assume that there exists a routing protocol that has already established multiple paths before the data transmission and that the base station and the sensor nodes have pre-built cryptographic algorithms and pre-distributed keys to generate the session keys before the deployment of WSN. They indicate that at the sending node level, each data packet should be split into N fragments and then reorganized into M groups where M is the number of cryptographic algorithms supported by the sending node. Each group is then encrypted with a different encryption algorithm, [13]. Each packet of the N encrypted fragments is encoded into $N + K$ fragments using the forward error correction (FEC) erasure coding before transmission. The obtained fragments are finally transmitted to the destination using multiple disjoint paths. After the reception of more than N encrypted fragments and their authentication and integrity checking, the receiving node reconstructs the encrypted data packet. The combination of data fragmentation over multiple paths combined with the multi-version encryption minimizes the chances of an attacker to get the complete data at one time, [13].

The scheme proposed in [13] can not be adopted for multihop relay networks as the routing architecture is a tree where each RS has only one superordinate. More precisely, it is not possible to transmit the packets of one flow on different disjoint routes because there exists only one branch (i.e., route) from the sending RS to the MR-BS. Another shortcoming of this protocol is that all the sensor nodes are involved in securing the transmission; this means that the base station has not to make an extra effort for guaranteeing security even if it is more powerful in terms of processing capabilities and power. Besides, this scheme makes it hard to ensure global QoS since fragments of the same packet will use different routes with different QoS capabilities and the destination needs to wait for getting the fragments; which may induce additional delays.

Tang et al. [14] designed two routing protocols that minimize the number of hops between a source node and a destination node while maximizing the lifetime of sensor networks. Besides, they considered security challenges targeting the wireless sensor networks and proposed a secure routing protocol called SecMLR which works in an energy-efficient way in order to resist potential attacks, [14]. In more detail, the adopted architecture combines wireless sensor networks and mesh networks by deploying mesh routers within the monitoring areas (i.e., sensor networks) and mesh gateways in order to transmit the

sensed data in long-distance and reliable fashion. The mesh gateways play the role of sink nodes, routers and gateways.

In order to optimize the performance of the routing service in the hybrid architecture, [14] proposed two routing protocols called shortest path routing (SPR) and maximal network lifetime routing (MLR) and then tried to secure the last one. The SPR protocol minimizes the number of hops of data transmission between each sensor node and a gateway; thus minimizing the total hops of a sensor network. However, the MLR protocol maximizes the network lifetime (i.e., the time when the first sensor node drains its energy) by minimizing the total energy consumption of all sensors in the network while minimizing the differences between individual nodes energy consumption and average energy consumption. Securing the MLR protocol by different measures gives birth to the secure maximal network lifetime routing (SecMLR). A node that needs to send data but does not have set up a routing table broadcasts a routing query to m destinations. When a node receives the request for the first time, it broadcasts the message after appending itself in the path field. Duplicate requests are not re-broadcasted. When a gateway receives a routing query packet, it verifies the authenticity of the sender and verifies whether the message is replayed by a malicious node by checking a counter value. After that, the gateway waits for a timeout in order to collect multiple path information and then calculates the shortest path between the source and the destination using a particular formula, [14]. Finally, a routing response is given back to the source. The gateways need to broadcast their new places using the μTesla protocol, [15], in order to achieve authenticated broadcast.

The fault or intrusion tolerance is simply achieved by setting up routing tables with multiple entries ending to specified gateways so that data which failed to reach the destination using a particular route may choose a different path. This scheme can be adopted for multihop relay networks only on the backbone between the neighboring intermediate BSs (i.e., from the sender MR-BS to the destination MR-BS). In more detail, the route from the sending RS to its MR-BS and the route between the destination MR-BS and its managed destination RS are already known and fixed (i.e., branches of one tree). Therefore, we do not have to choose from different paths. Besides, this scheme was designed in order to save the energy of the sensing nodes adopting it. In multihop relay networks, the MR-BSs which are the unique entities able to adopt this scheme do not have energy constraints. Last but not least, authors of [14] did not consider particular QoS constraints in terms of delay, jitter or throughput in their proposed secure routing protocol. Therefore, applying this scheme for multihop relay networks will not provide us with the required QoS guarantees.

To the best of authors' knowledge, there is presently no research work that addresses the issue of intrusion tolerance within relay networks. This is mainly due to the fact that the proposed intrusion tolerance solutions for ad hoc, sensor or mesh networks can not be applied to the relay context without introducing some modifications induced by the particularities of the relaying concept. More precisely, the network topology of the multihop relay is a tree rooted at the MR-BS which differs from the mesh topology of ad hoc, sensor or mesh networks. Second, the Multihop Relay mode is simpler than the ad hoc or mesh mode as mobile nodes within a relay network just enjoy the connection while all the processing regarding the relaying activity is transferred at the Relay Station level. Moreover, the relay mode is intended to be more stable than the ad hoc, sensor or mesh ones because the MR nodes relaying traffic and signalling have more computing and power capabilities and are managed by the network operator.

2.3 Requirements for an efficient QoS provision over MMR networks

The IEEE 802.16j amendments introduce multiple concepts related to QoS provision such as service flow QoS scheduling, dynamic service establishment and two-phase activation model, [1]. Five data delivery services are defined which are the unsolicited grant service (UGS), the real-time variable-rate service (RT-VR), the non-real-time variable-rate service (NRT-VR), the best effort service (BE) and the Extended real-time variable-rate service (ERT-VR). These services support real-time applications generating fixed and variable data rates along with applications requiring a guaranteed data rate while being insensitive to delays and applications with no rate or delay requirements. Nevertheless, the IEEE 802.16j amendments do not indicate how to implement such services. Although the amendments identify QoS parameters such as maximum latency, tolerated jitter, minimum reserved traffic rate, maximum sustained traffic rate and traffic priority that implement the provided services, the choice of a specific algorithm or procedure that implements the QoS provision and optimizes it is left to the network operators, [1].

Besides, the IEEE 802.16j amendments define signalling messages that are used when discovering routes with a predefined QoS level but do not specify an algorithm that manages the routes establishment and maintenance in case of a handover while providing the required QoS level. Therefore, specific call admission control (CAC) algorithms and route maintenance in case of mobility algorithms should be defined by the network operators with regard to their specific needs. Meanwhile, the IEEE

802.16j amendments do not indicate particular procedures that enable the QoS estimation over the links. For instance, it is not indicated how to evaluate the delay or the throughput value over a wireless link. Besides, it is not indicated how to verify the pretended QoS values over the wireless links.

If security is considered as an additional QoS parameter, the IEEE 802.16j amendments do not specify particular procedures that enable the detection of potential attacks conducted by malicious SSs and probably malicious RSs. Therefore, a successful timely-based attack or denial of service (DoS) or distributed DoS (DDoS) attack against the MMR network may degrade the QoS requested by legitimate customers and impair the corporate image of the network operator.

In order to achieve an efficient QoS provision, the MMR networks should implement efficient algorithms for CAC that enlighten the choice of the best routes affording the required QoS level. Besides, specific procedures should enable the route maintenance in case of RSs or SSs mobility so that the customers can enjoy ubiquitous services with nearly the same QoS level despite their high mobility. Last but not least, the MMR network should be intrusion tolerant in order to continue providing the agreed QoS level despite intrusions. In more detail, an intrusion tolerant MMR network will enhance the QoS provision through using specific ways to detect potential intrusions and efficiently reacting to them without dramatically altering the QoS level agreed with the customer.

In this article, we propose an architecture called 3TCA for MMR networks. The 3TCA architecture enables a trusted QoS estimation over wireless links while compensating for the impact of mobile RSs handoff on the affected flows. Thanks to the 3TCACs, the proposed architecture achieves also timely-based and DoS attacks detection and continues delivering the nearly same QoS level despite such attacks. A comparison of the 3TCA architecture and the existing research work is given in Table 1.

3 3TCA: a novel trusted timely attacks-tolerating communication architecture

We define in this section our novel trusted timely attacks-tolerating communication architecture called 3TCA and we describe the services that it guarantees in order to achieve QoS and intrusion tolerance provision for MMR networks.

Table 1 Comparing the 3TCA architecture with the current research work

Research work	Characteristics	Shortcomings
Self optimization HandOver (HO) mechanism,[4]	Reduces HO numbers and optimizes channel scanning	-Does not consider security issues-Uses GPS
Fast HO scheme, [5]	Minimizes collisions in order to achieve fast HO	Does not consider security issues
Speed sensitive HO scheme, [6]	Speed sensitive HO under hierarchical cellular system	Does not consider security issues
Path selection method, [7]	Secures the routing for MMR networks	Does not consider QoS issues
QITAR, [8]	Achieves QoS and intrusion tolerance for ad hoc networks	-Vulnerable to high-speed mobility-May experience scalability problems
MERQIT, [9]	Achieves QoS and intrusion tolerance for mesh networks	-Clusterheads are mobile nodes that may experience attacks and leave the network-Isolates malicious nodes
WSN-TUM, [13]	Fault-intrusion tolerant framework for sensor networks	-Split the packet on fragments and send them on disjoint routes. However MMR networks use the same route as the routing architecture is a tree rooted at the MR-BS-Global QoS is hard to guarantee since each fragment uses a different route with different QoS constraints-All sensors participate in the securization effort. For MMR networks, this should be the task of the MR-BS
SPR, MLR and SecMLR, [14]	Secure routing protocols for sensor networks	-Intrusion tolerance is simply achieved by setting up routing tables with multiple entries to specified gateways-If the scheme is adopted for MMR networks, it can be only implemented on the backbone-QoS constraints are not considered
Our method	-Architecture designed for MMR networks-Achieves QoS and intrusion tolerance for MMR networks-Compensates HO delays-Detects attacks and compensates their impact on the agreed QoS	

3.1 3TCA definition

The 3TCA architecture is formed by the interconnection of trusted software components called 3TCA components (3TCACs). The 3TCACs are implemented on the access nodes which are the RSs and the MR-BSs and form a backbone allowing fast and protected communication between its nodes. The interconnection of 3TCACs within the relay network gives birth to a distributed component that provides trusted time-related and secure-related services along with an intrusion-related service that guarantee intrusion detection and tolerance to a set of attacks. Meanwhile, the 3TCACs provide QoS compensation in order to minimize the negative impact of handover and possible attacks on the provided QoS. The interconnection of 3TCACs is assumed to be accomplished via a completely secure wireless channel that implements cryptographic tunnels as illustrated by Figure 2.

The 3TCA architecture also defines 3TCA Entities (3TCAEs) which are application-level entities that reside over the Operating System and communicate via insecure wireless channels. 3TCAEs may be attacked and behave maliciously. The 3TCAEs use the trusted services of the 3TCACs in order to participate in the 3TCA' protocols implementation within an intrusion-tolerant environment.

Thanks to the 3TCACs, the 3TCA architecture enables the provision of trusted services and the correct implementation of the 3TCA protocols despite the probably malicious behavior of the 3TCAEs and the attacks that may target the wireless insecure channel. Since all the 3TCACs are synchronous while 3TCAEs are asynchronous and since all 3TCACs are assumed to be secure while 3TCAEs may be malicious, we conclude that the 3TCA architecture provides a hybrid intrusion-controlled environment with distributed trusted components. The 3TCACs implement the necessary validations and verifications to provide trusted values of QoS, detect the possibly malicious behaviors and most importantly compensate the negative impacts of group mobility and some time-related and DoS attacks.

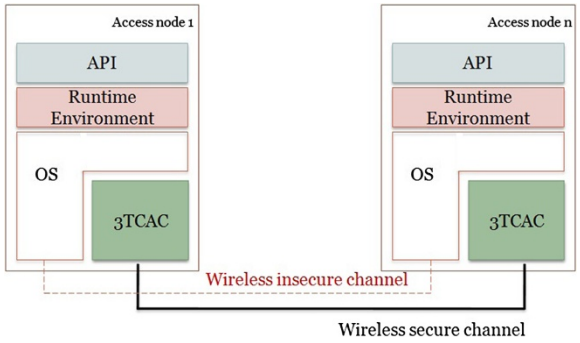

Figure 2 Interconnection of 3TCACs.

3.2 3TCA services

The 3TCA services are provided by the trusted 3TCACs and may be summarized as follows:

3.2.1 Trusted time-related services

The 3TCACs provide two trusted time-related services, namely the trusted absolute timestamping service and the trusted duration measurement services.

- The trusted absolute timestamping service: provides a global timestamp value shared between all 3TCACs as each 3TCAC is equipped with an internal clock that is synchronized with all other 3TCACs' clocks. The precision of the timestamp value is determined by the precision of the adopted clock synchronization protocol. Note that the 3TCAE entity which asks its 3TCAC for a trusted timestamp value will receive it after a variable delay that highly depends on the processing capabilities of the hosting node.
- The trusted duration measurement service: computes the time required to execute a function by evaluating the beginning and the ending timestamps of that function.

3.2.2 Trusted security-related services

The trusted authentication service and the trusted random number generation service are the security services provided by the 3TCACs.

- The trusted authentication service: enables a 3TCAE to authenticate itself and to communicate securely with its local 3TCAC. We assume that each 3TCAC has an asymmetric key pair. The 3TCAE authenticates itself to its local 3TCAC through sending a challenge X_e and a symmetric key K_e both encrypted with the 3TCAC's public key. The 3TCAC authenticates itself to the 3TCAE entity by sending the signature of the challenge encrypted with its private key. After that, the symmetric key K_e is used to encrypt the exchanged data between the entity 3TCAE and its trusted component 3TCAC.
- The trusted random number generation service: generates trustworthy uniformly distributed numbers that are required for building cryptographic primitives such as authentication protocols.

3.2.3 Trusted intrusion-related service

The novel trusted thresholds handling service described below is also provided by the 3TCACs.

- The trusted thresholds handling service: manages the variables configured with threshold values and generates alert messages when the thresholds are exceeded.

3.3 3TCA compensation

The 3TCA architecture provides QoS compensation for handing over RSs by accelerating the affected flows. More precisely, once the duration of the handover process is computed by the 3TCACs, the QoS of the affected flows is revised through decreasing the delay values and optionally decreasing the jitter values and increasing the bandwidth by adopting either a linear approach or an exponential approach. A new CAC is processed with the updated QoS values on chosen links in order to rapidly find a suitable route.

On the other hand, malicious RSs may try various time-related and DoS attacks. Unfortunately, isolating such malicious actors will dramatically decrease the coverage area since the relay topology is a tree and the isolation of one RS may doom a whole branch. To address this issue, the MR-BS managing the malicious RS will calculate the delay and/or jitter induced by its presence and then it will compensate it according to the load of the involved links. The compensation is fulfilled through executing a new CAC with updated values of delay, jitter and throughput. Note that the MR-BS may send a different signalling message containing different delay, throughput or jitter values in order to individually perform admission control on the links forming the path to the destination as each link may have a different load condition. However, the MR-BS will choose the values for CAC so that the end-to-end QoS requirements are met even if each link on the path may provide a different QoS.

The trusted timely computing base (TTCB) is a software component which has been developed in the malicious and accidental fault tolerance for internet applications (MAFTIA) project. The main goal of the MAFTIA project is to construct dependable trustworthy applications implemented by a set of distributed components with varying degrees of trustworthiness, [3]. The TTCB module can be viewed as a secure real-time distributed component that guarantees trusted services related to time and security such as the trusted block agreement, the trusted duration measurement, and the trusted absolute timestamping, [16]. In an Internet architecture, each host has a local module called the *local TTCB*. These modules are interconnected by the secure control channel and form the distributed trusted component (DTC). The TTCB assists the applications running between participants in the concerned hosts which are interconnected by a vulnerable Payload Channel and form the payload system subject to arbitrary byzantine failures. Nevertheless, the TTCB modules do not handle threshold values in order to detect malicious behavior and react in time. Moreover, the TTCB modules were not programmed to perform QoS compensation.

3.4 3TCA detailed description

The 3TCA architecture is a novel architecture designed for MMR networks which provides QoS and intrusion-tolerance despite group mobility and probable attacks that may target the access nodes (i.e., RSs and MR-BSs). This architecture is based on trusted 3TCACs which offer time-related, security-related and intrusion-related services in order to form a secure communication environment with QoS guarantees. QoS provision is achieved first through securely estimating the QoS parameters using the trusted 3TCACs services and second through compensating the negative impact of handover and a set of attacks on the previously agreed QoS level. The TTCB component defined in the frame of the MAFTIA project lacks threshold management and compensation processing and offers useless services (i.e., such as trusted agreement) for the relay context; therefore, it needs to be revised in order to be adopted within the 3TCA architecture.

The 3TCACs are enriched with tables and matrices describing the network entities so that they become able to control their behavior. Besides, some threshold variables were included so that the 3TCAC component can decide whether an entity is malicious. Furthermore, we made the 3TCAC component the sole entity that creates the control messages, signs or encrypt them (i.e., sender's 3TCAC) and then verify and process them (i.e., receiver's 3TCAC). Last but not least, we used the time related services to make the 3TCAC component detect timely-based attacks and compensate the impact of both handover and malicious behaviors.

Moreover, the 3TCACs have been enhanced in order to provide intrusion tolerance for multihop relay networks while respecting the QoS constraints. Even if malicious entities try various attacks, their potential damage will be limited since routing actions will be globally processed and secured by the 3TCACs while a timed behavior is supported to provide QoS guarantees. The 3TCACs are deployed on each RS and MR-BS. Each 3TCAC is equipped with a matrix encompassing the registered SSs and subordinate RSs along with the QoS and intrusion information related to them. The signalling messages are created and processed by the 3TCACs in order to validate the contained information and detect the entities trying to compromise the integrity or the identity of the issuer.

We also defined threshold values for delay and jitter that may be dynamically updated in order to detect the entities that refuse to participate in the QoS routing process through pretending higher delay and jitter values. The *probably-malicious-RS* variable is incremented each time the 3TCAC notices an infraction. When the *probably-malicious-RS* value reaches a pre-configured threshold, a *probably-malicious-RS-Alert*

message is forwarded until the MR-BS. Upon receiving that alert, the 3TCAC of the MR-BS will try to compensate the QoS disturbance caused by the malicious behavior through updating the QoS values on some chosen links.

Furthermore, the 3TCACs which are synchronized will raise an infraction if they do not receive the signalling messages within a pre-configured threshold period. In this case, they will increment the *probably-malicious-RS* variable and send a *probably-malicious-RS-Alert* message when required. Multiple colluding SSs may decide to generate simultaneous requests to cause a distributed DoS to the managing RS. To address this issue, the 3TCAC kernels will be configured to not process more than a threshold value of requests sent by the same SS within a period of time and to not process more than n requests or QoS estimations overall.

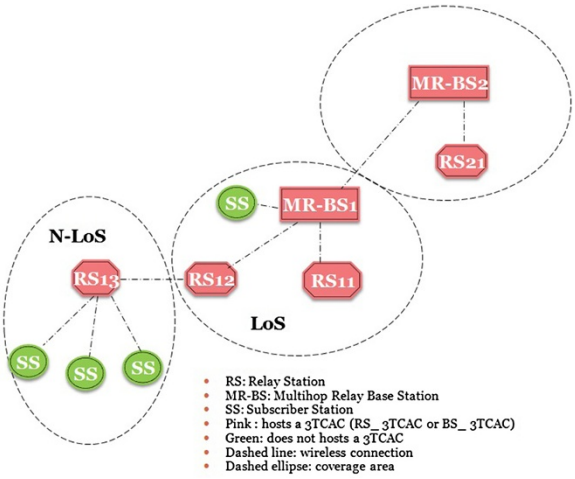

Figure 3 The adopted multihop relay network model.

4 3TCA implementation in relay networks

We propose in this article an architecture called 3TCA that provides intrusion tolerant routing and QoS for MMR networks. In more detail, we propose to guarantee QoS in terms of delay, bandwidth and jitter for the multihop relay networks while addressing mobility through performing handover disturbance compensation in case of mobile RSs. Meanwhile, we tolerate particular timely-based and DoS attacks through compensating their impact on the agreed QoS level.

4.1 Network modeling

The multihop relay networks are organized in a tree topology and define the RS stations which relay the traffic of N-LoS SSs to the MR-BSs. MR-BSs manage the subscribers of their coverage areas and enable the communication with external networks. The assumed multihop relay network depicted in Figure 3 accommodates three kinds of nodes. The relay stations may be fixed or mobile. The RSs and the MR-BSs host 3TCACs while SSs do not. The hosted 3TCACs will be respectively called RS_3TCAC and BS_3TCAC. Non Line of Sight SSs (NLoS SSs) connect to the MR-BSs through the RSs. However Line of Sight SSs (LoS SSs) may directly be managed by MR-BSs. Viewed by a NLoS SS, the network topology is tree-rooted at the MR-BS where a branch is formed by intermediate RSs. We also assume that a mobile RS (i.e., always found in trains and other transportation systems) can only be a superordinate access station for the passengers' SSs moving with the vehicle and that it can only have fixed RSs as superordinates. We argue that having mobile RSs as superordinates for non-passengers' SSs wastes time and resources as the mobile superordinate may move away from its subordinate either during the management phase (i.e., network entry, path establishment, etc) or during the data transfer phase. In this case, the managed subordinate should reprocess the management procedure or wait until a new path is found. Therefore, we always consider a fixed network backbone while the mobile RSs will only manage mobile SSs that are moving with them.

4.2 QoS provision within the multihop relay networks

We design an architecture called 3TCA that provides QoS in a trusted way. Mobility will be particularly addressed in order to compensate the delays induced by handing over RSs. Besides, timely-based attacks will be tolerated while respecting the particularities of the multihop relay networks.

4.2.1 QoS Estimation over routes linking fixed entities

In our network model, the MR-BSs and the fixed RSs form a fixed backbone and each MR-BS, RS and SS should be uniquely identified within the network. We assume that each fixed RS knows exactly its superordinate on the path to the MR-BS. Such information may be assigned by the operator or provided by the managing MR-BS. The MR-BS assigns a unique path-ID to each branch and each fixed RS records the path-ID(s) of the path(s) it belongs to. We propose that each fixed RS periodically initiates a QoS estimation over the link between it and its superordinate. The estimated QoS values in terms of Delay (ms), Rate (bits/s) and Jitter (ms) are sent to the superordinate within a *BS-QoS-INFO* message and should be forwarded until the MR-BS.

After collecting the QoS information of each link, the MR-BS deduces the global QoS that can be guaranteed over the whole path (branch) as the maximum value of delays over the links forming the path, the minimum values of rates over the links forming the path and the maximum value of jitters over the links forming the path. Besides, each MR-BS periodically estimates the QoS over the wireless links between it and its neighboring MR-BSs.

The period value may be fixed by the network operator. Note that the QoS estimation procedure should be initiated from the leaf fixed RS to the root MR-BS in order to minimize the number of transmitted signalling messages (i.e., overhead).

4.2.2 Admission control of handing over mobile relays

When a mobile RS enters a new coverage area, the mobile RS needs to serve multiple service flows issued by its managed SSs. A separate admission control procedure should be executed for each service flow. We think that the available QoS between the managed SSs and the mobile RS needs to be re-estimated since the physical properties of the wireless link may change. The mobile RS needs to adapt the provided QoS in order to compensate the handover disturbance and the probable modification of the available QoS on the edge links (i.e., links between it and the managed SSs). For that reason, we propose that each mobile RS entering a new coverage area initiates an edge QoS re-estimation with each managed SS and then adapts the QoS requirements of each flow issued by the concerned managed SS. After that, the mobile RS sends a DSA-REQ on its name to its superordinate for each flow. That request encompasses the source identifier, the destination identifier and the required QoS in terms of minimum rate, maximum delay and tolerated jitter. Note that in our case, the DSA-REQ will travel from a subordinate mobile RS to its superordinate in order to minimize the overhead.

In our model, the MR-BS already has an estimate of the QoS on the sub-path from the mobile RS to it, the MR-BS can also receive QoS information of other paths on the backbone. The MR-BS may directly route flows which can not tolerate delays using the QoS information of the candidate paths. The MR-BS may also send a DSA-REQ to all the RSs on the candidate path(s) in order to request an admission control decision as specified by the IEEE 802.16j amendments because the QoS information can be updated, especially when the configured period of QoS re-estimation is relatively high.

4.2.3 QoS estimation

In this section, we detail the procedures adopted in order to estimate the QoS parameters in a reliable manner using the 3TCAC modules. We also overview the mechanisms that will be used to fulfill handover disturbance compensation.

If a sequence of packets is sent from a source point A to a destination point B, each of the packets will need a slightly different time to reach the destination. Formally, jitter may be defined as the statistical variance of the data packet inter-arrival time. According to [17], and when adopting the real time protocol (RTP), jitter is measured in timestamp units. For example, if the transmitted audio is sampled at the usual 8000 Hz, the unit will be $\frac{1}{8000}$ of a second. The endpoint computes an estimate using a simplified formula (i.e., a first order estimator). More precisely, to estimate the jitter $J(i)$ after the reception of the $i$thpacket, we need to calculate the change of the inter-arrival time, and then divide it by 16 to reduce noise and add it to the previous jitter value as described by the following formula $J(i) = J(i-1) + \frac{(|D(i-1,i)|-J(i-1))}{16}$, where the value $D(i-1,i)$ is the difference of relative transit times for the two packets. That difference is computed as $D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$ where $S_i$is the timestamp from the packet i and $R_i$is the time of arrival for packet i, [17]. The assumption which is made is that the sender sends one packet each 20 milliseconds and that the ideal transit time is 10 milliseconds. The computation also starts from zero and not from a random value, which means that $S_i = 0$ and $R_i = 10$. The jitter value starts to grow slowly despite large differences. In fact, when the large differences disappear (i.e., $i > 8$), the estimate starts to approach the approximate mean value, [17].

We estimate the jitter value using the 3TCAC related services which are the trusted absolute timestamping service and the trusted duration measurement service. The QoS values will be estimated only on the insecure wireless channel. In order to estimate the jitter in a reliable fashion on the backbone, we propose that 3TCAE($n-1$) calls its RS-3TCAC($n-1$) each 20 ms in order to get a QoS-Estim message and then send it to 3TCAE(n) on the insecure wireless channel as shown in Figure 4. 3TCAE(n) forwards the received message to its 3TCAC which verifies the QoS-Estim and then extracts the valuable information required to compute the jitter. That pre-described procedure is repeated 9 times in order to estimate a mean jitter value. If the computed jitter value does not exceed a pre-configured *MaxJitterThreshold*, 3TCAC(n) will add it to a BS-QoS-INFO message that will be given back to the application entity in order to forward it to the superordinate.

In order to estimate the jitter on the wireless edge links between the RS and the managed SSs, the RS-3TCAC sends a QoS-REQ signalling message to its payload entity and then triggers the trusted absolute timestamping service. The 3TCAE entity forwards the received packet to the SS which should answer with a first QoS-RSP message. 3TCAE receives the QoS-RSP and then forwards it to the RS-3TCAC which computes the edge delay between the RS and the SS. As the jitter computation requires a synchronization between the entity which will send the equivalent of the RTP packets each 20 ms and the entity which will receive them, the RS-3TCAC and the SS need to have the same clock values.

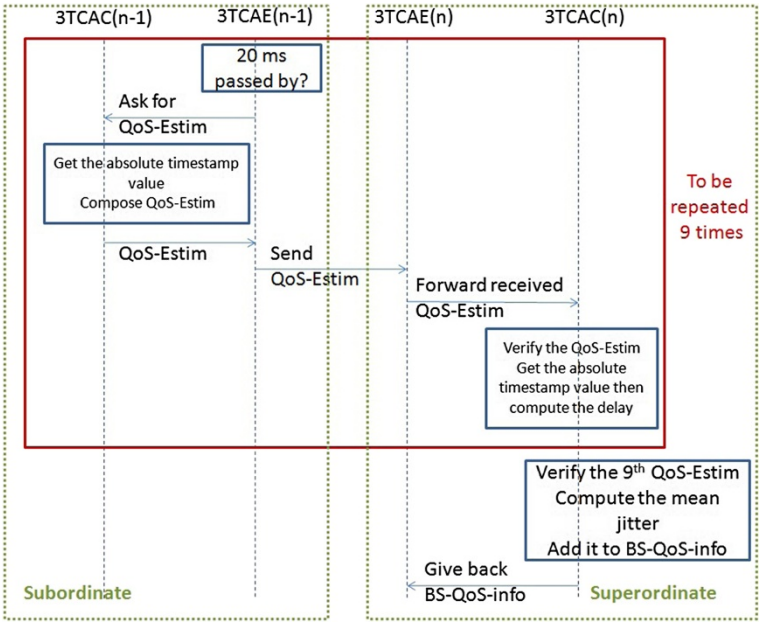

Figure 4 **Trusted jitter estimation on the backbone.**

However, we already stated that only 3TCAC modules are synchronized. Therefore, the SSs may have completely different clock values compared to their managing RSs. To address this issue, we propose to use the already computed delay value in order to synchronize the SS and its managing RS. In more detail, on receiving the first QoS-RSP message, the RS-3TCAC generates a QoS-Sync message and then sends it to its application entity.

The QoS-Sync message indicates the clock value that should be adopted by the SS when sending the next 12 QoS-RSP messages. That clock value is given by the following formula $C_0 = currentTimestamp + \alpha +$ computed edge delay, where *currentTimestamp* is the value of the timestamp when computing the clock value and α is the delay of the communication between the 3TCAC and its entity added to the delay required by the receiving SS to process the incoming message added to an estimate of the possible error. When receiving the QoS-Sync message, the SS synchronizes its clock and then sends 12 consecutive QoS-RSPs to the 3TCAE entity of the managing RS-3TCAC . We propose to use only 9 received packets in total in order to estimate the mean jitter value on the edge link; the 3 remaining packets are sent in order to tolerate possible packets loss. After getting 9 packets, the RS-3TCAC ignores the other packets. If the jitter estimation fails, it will be resumed after a configured threshold period in order to tolerate the behavior of malicious entities which try to conduct a DoS by over-exploiting the RS's resources.

The delay estimation on the wireless edge links and on the backbone is very similar to the jitter estimation one. More precisely, the delay will be estimated using the same control packets required to compute the jitter. Note that the mobile RS should re-estimate the delay on the edge links in case of handover. Besides, the requested delays for the served flows will be updated in order to compensate the handover delays.

4.2.4 *Handover disturbance compensation*

The handover process induces delays mainly caused by the signalling exchange and the switching of the connection to a new access station. Therefore, the additional delays should be compensated in order to provide the level of QoS that was agreed. Besides, we assume that the handing over RS does not start the compensation procedure before performing handover and allowing each managed SS to perform the QoS re-estimation. We propose that when the handing over mobile RS begins the handover process, it triggers the trusted duration measurement service. When the mobile RS attaches to a new access node and the managed SSs terminate the QoS re-estimation procedure, the trusted duration measurement should be stopped in order to compute a trusted value of the duration of the handover process. The obtained delay is then subtracted from the delay values of the current flows if its value is smaller than the previously agreed value. However, if this is not the case or if we may not find an available route that offers the updated value, we may try to subtract a portion of the handover delay using either a linear

approach or an exponential approach. Note that regarding the first attempt of QoS compensation, the MR-BS managing the handing over RS may select a route based on the stored values of QoS offered by the routes to the destination. Nevertheless, that first attempt of QoS compensation may fail because the MR-BS may take its decision before the periodic update of the offered QoS.

Regarding the linear approach, we assume that a mobile q x milliseconds to execute the handover process. We also assume that we updated the delay value by subtracting the delay needed for handover and that we performed a first CAC on a selected route and that CAC was fruitless. The 3TCAC component of the handing over RS triggers the trusted duration measurement service in order to compute the delay of the first admission control process. Let that delay be equal to del. We propose to launch 8 parallel admission control procedures, first we update the delay value by subtracting $x + \text{del} - 8y$, and then we subtract $x + \text{del} - 7y$ and so on until we find a suitable route that supports an updated delay value. the choice of the value of y may be random or it may be proportional to the value of x; it will vary according to the environment conditions and the load. We may then sense the satisfaction degree of the user when the compensation is processed and the satisfaction degree of the user when the compensation is not processed. The handover disturbance compensation using the linear approach in case where the handover delay is smaller than the initial flow's delay and where the first processed CAC is fruitless is described by the pseudo-code in Algorithm 1.

The choice of executing 8 parallel admission control procedures is totally random, other numerical scenarios may be experimented. We remind that the choice of the value of y may be random or it may be proportional to the value of x; it will vary according to the environment conditions and the load. Nevertheless, the value of $x+\text{del}-y$ needs to be inferior to x. Regarding the exponential approach, we propose to launch 4 parallel admission control procedures, first we update the delay value by subtracting $x + \text{del} - 8y$, and then we subtract $x + \text{del} - 4y$, and then we subtract $x + \text{del} - 2y$ and finally we subtract $x + \text{del} - y$ until we find a suitable route that supports an updated delay value.

As the jitter is the variation in time between packets arriving, accelerating packets through the network will reduce the jitter. More precisely, the jitter caused by handover may be compensated through determining when packets should arrive at the destination and then accelerating them accordingly. Moreover, the handover disturbance compensation in terms of delay or jitter may also be achieved through augmenting the value of the minimum required rate since providing more bandwidth to a flow results in accelerating that flow. Besides, the handover compensation in terms of bandwidth may be combined to the handover compensation in terms of delay in order to minimize the handover impact on the QoS sensitive flows and increase the probability of finding a suitable new route that fulfills the new QoS constraints.

Algorithm 1 Handover compensation using the linear approach.

If (RS begins handover)
Start Trusted Duration Measurement Service
If (RS ends handover)
End Trusted Duration Measurement Service
Compute the handover delay = x (ms)
Initial flow delay = ifd
If ($x >$ ifd)
{ifd = ifd $-x$
Initiate CAC on chosen link
Start Trusted Duration Measurement Service
If (CAC on chosen link ended)
End Trusted Duration Measurement Service
Compute the CAC delay = del (ms)
If (CAC is fruitless)
// Process linear compensation
Initiate y
For ($i = 0$; $i < 8$; $i++$)
{Choose a different link
Launch CAC on chosen link with updated delay value = $x + \text{del} - (i^*y)$
Wait for the corresponding result
}
If (one suitable route is found)
Send flow on the suitable route
}*// end If (x > ifd)*

4.3 Intrusion tolerance within the multihop relay networks

The intrusion tolerance concept is implemented through performing the intrusion detection and then implementing mechanisms in order to survive the detected intrusion. In an IEEE 802.16j context, the RSs are normally controlled by the operator; thus, the probability of attacking them is much smaller than the probability of attacks within a traditional mesh network. Even if some attacks, such as selfish behavior, are not very common because the RSs do not have their own flows to transmit, an attacker may take control of one or more RSs or place a rogue RS, [18]. In order to fulfill the intrusion tolerance property, we enable the sub/superordinate RS or the MR-BS to detect the malicious party and then act.

4.3.1 Intrusion tolerance context

The relay RSs within multihop relay networks are controlled by the operator. Therefore, the probability of taking

control over RSs is relatively low. We propose to only use the insecure wireless channel to forward the signalling messages within multihop relay networks but we use the 3TCAC services to detect some attacks. Using only the insecure wireless channel enhances the scalability property of our protocol and reduces the overhead.

In order to protect the integrity of the signalling messages and verify the identity of the sender, each 3TCAE(n) forwards the received message to its 3TCAC(n) which uses encryption keys for un-signing and verifying it. Besides, all the signalling messages are produced by the 3TCAC and signed by it then relayed to the entity. Therefore, we are sure about the validity of the control information composing the message.

4.3.2 Tolerating time-based attacks

Let us assume that the entity of the RS (subordinate 3TCAE($n-1$) or current 3TCAE(n)) does not send the control message at time. Two principal causes may be responsible of this incident. The first cause is that the processing capacities of the RS(n) or of RS($n-1$) are saturated. The second cause is that either RS(n), or RS($n-1$), or both have been controlled by a malicious party and that it has become a malicious RS. If the RS becomes malicious, the delay and/or jitter will seem much longer on the route; thus probably causing DoS to managed SSs. The managed SSs will probably hand over because they can not be correctly served. If multiple malicious RSs are injected within a certain region/area of the network, a distributed DoS occurs.

In order to tolerate the previously described attack, we define two threshold values: the *MaxDelayThreshold* and the *MaxJitterThresold*. When the RS-3TCAC(n) computes the jitter or the delay on the link between itself and its subordinate, it compares the obtained values to the threshold ones. If the obtained values are larger than the thresholds, we propose that RS-3TCAC(n) renews its entity and then increments the *probably-malicious-RS* variable for it and its subordinate. When the *probably-malicious-RS* variable reaches a pre-configured threshold, the concerned 3TCAC sends a *probably-malicious-RS-Alert* signaling message which will be forwarded till the MR-BS. The values of *MaxDelayThreshold* and *MaxJitterThresold* may be dynamically updated according to numerous factors such as the history of the feasible QoS on the links, the load conditions, etc.

When 3TCAE($n-1$) asks its 3TCAC($n-1$) for providing the QoS-Estim message that will be used for jitter estimation, 3TCAC($n-1$) will verify that it is really receiving the demand for that signalling message every 20 ms using its trusted absolute timestamping service. However, note that 3TCAC($n-1$) can not verify whether 3TCAE($n-1$) has really sent the QoS-Estim message as soon as it (i.e., that entity) received the demanded message.

The trivial idea to prevent the malicious RSs from succeeding in their attacks is that the MR-BS isolates them. However, isolating the malicious RSs will decrease the coverage area. More precisely, the more the isolated malicious RS is close to the root MR-BS within the network tree, the more the probability of deleting multiple routes increases and the more the coverage area of the network is minimized. Besides, even if the isolated malicious RS is a leaf in the network tree, all the SSs served by it need to change their access entity and a whole area will become uncovered. Nevertheless, only the route comprising that malicious leaf RS will be deleted and multiple other routes will not be affected by the isolation.

We think that isolating the malicious RSs will have severe consequences on the performance of the MR network. If the previously described time-based attack occurs when evaluating the QoS for the first time, the MR-BS may temporarily adopt the estimated values (i.e., which are worse than the values that would be estimated if the RS was not malicious) and then trigger a maintenance procedure that will be handled by the operator's control center. However, if the detection of a malicious RS occurs during the data transmission phase, the MR-BS which receives the *probably-malicious-RS-Alert* signaling message has already stored the QoS that can be provided on each link of the concerned route. In order to tolerate the attack, the MR-BS calculates the delay and/or jitter induced by the presence of the malicious RS by comparing the new QoS values enclosed in the *probably-malicious-RS-Alert* signaling message to the old QoS values which were used to admit the flows. Then, the MR-BS chooses the links (i.e., RSs) that will be involved in the compensation procedure according to the load and sends them a DSC-REQ signaling message while indicating the new values of delay, jitter and throughput. The MR-BS also triggers the maintenance procedure in order to rapidly fix the cause of the malicious behavior of the attacking RS.

4.3.3 Tolerating attacks caused by the SSs and attacks on the backbone

SSs are mobile entities which can either directly attach to the MR-BSs or access the MR network via RSs. Colluding SSs acting within the same coverage area may try to cause a DoS to their access point by sending multiple requests at the same time in order to saturate the processing capabilities of the managing RS or MR-BS. In order to tolerate such attack, the 3TCAC kernels will be configured to not process more than a threshold value of requests sent by the same SS within a period of time and to not process more than n requests or QoS estimations overall.

In order to tolerate replay attacks on the backbone, we indicate that each 3TCAC which composes a new

signalling message should append to it the current timestamp value. As all 3TCAC modules are synchronized, the replay attacks attempted by malicious entities will be easily detected when the receiving 3TCAC verifies whether its current timestamp value is much superior to the value indicated within the received signalling message added to the delay on that link.

4.3.4 Tolerating RSs refusing to forward signalling messages on the backbone

Malicious RSs may refuse to participate in the periodic QoS estimation procedure or may refuse to forward the signalling messages such as QoS-Estim, BS-QoS-INFO, etc in order to prevent an efficient management of the QoS. If they collude with other malicious RSs, they may refuse to forward the *probably-malicious-RS-Alert* in order to prevent the MR-BS from adapting the QoS disturbance compensation and triggering the maintenance procedure. In order to tolerate the attacks targeting QoS estimation, each 3TCAC(n) module should verify whether it is receiving the signalling message from 3TCAC($n-1$) within a tolerance period. If it is not the case, 3TCAC(n) will increment its own *probably-malicious-RS* variable and the *probably-malicious-RS* variable of its subordinate and send the *probably-malicious-RS-Alert*describing the RS for which the*probably-malicious-RS* variable has reached the configured threshold.

The tolerance period is computed as the period of estimating QoS added to the previously-computed delay on the link added to a maximum threshold value. Colluding malicious RSs which refuse to forward the *probably-malicious-RS-Alert* are not easily tolerated as each RS can only have one superordinate. In more detail, if the RS(n)'s superordinate and subordinate are both colluding malicious RSs, the*probably-malicious-RS-Alert* sent by RS(n) in order to inform about the malicious behavior of the subordinate will never reach the MR-BS and RS(n) will not be able to detect that fact.

4.3.5 Tolerating RSs refusing to forward data flows

We already specified that the MR-BS stores the QoS information related to the routes but that it may perform admission control for certain flows. Note that certain malicious RSs may indicate negative responses to the received DSA-REQ. To tolerate such an attack, the managing MR-BS should keep the history of the negative responses. As the QoS estimation is periodically processed, a malicious RS that tries to pretend higher delay and jitter values will be easily detected by its 3TCAC as it has been described earlier (i.e., detecting time-based attacks). In this case, the fact that the RS negatively responds to the DSA-REQs may confirm that it has a malicious behavior. However, if the RS is malicious but it has correctly performed the QoS estimation within the next period, the value contained in the *probably-malicious-RS* table will be incremented. Besides, if that value reaches a threshold, the MR-BS will order the SSs which are managed by the concerned RS to hand over and then trigger the maintenance phase.

A malicious RS may try to cause DoS for some of its managed SSs by refusing to relay their flows and by stopping relaying the MR-BS messages to them while updating the list of its managed SSs. The 3TCAC of the malicious RS will no longer receive data from the attacked SSs. The MR-BS deduces that the attacked SSs have been shutted down because if they were alive and they quited the coverage area, they should have initiated a handover process. This attack is more severe if the malicious RS is a mobile one because the IEEE 802.16j specifications indicate that the managed SSs of a mobile RS need to handover with it (i.e., follow it) so that the attacked SSs can not independently perform handover or network re-entry. This attack is not easy to counter because the managed SSs may suddenly shut down in case of power shortage for example. When the 3TCAC of an RS can no longer receive messages from a threshold number of SSs, the *probably-malicious-RS* variable should be incremented. When that variable reaches a threshold value, the *probably-malicious-RS-Alert* will be sent to the 3TCAC kernel of the managing MR-BS over the secure wireless channel in order to trigger the maintenance procedure.

5 Properties and features of 3TCA-based relay networks

The 3TCAC architecture for multihop relay networks compensates the QoS level of handing over SSs through accelerating the affected flows on some links of the established path. More precisely, the delay of the handover operation is first measured by the handing over RS. After that, a compensation of that delay is processed through increasing the bandwidth values and decreasing the delay and jitter values relative to the affected flows on chosen links along the route. To that end, a first CAC is processed in order to determine the links that may accelerate the affected flows. If that CAC is not successful, a second compensation is attempted while adopting a linear or an exponential approach in order to compensate a portion of the handover and the first CAC induced delays.

In order to authenticate the SSs and protect the signalling messages exchanged between the SS and its access entity, each SS shares a private key with the managing RS or MR-BS. Colluding malicious SSs which send simultaneous requests to the managing RS will not succeed in causing a distributed DoS as the 3TCACs process the incoming messages and are configured to not handle more than n requests overall.

Malicious RSs may try to replay the control messages in order to cause a DoS to their superordinate or to tamper with the results of the QoS estimation. The 3TCACs will resist to the DoS because they are configured to not process more than n signalling messages overall. Besides, each signalling message encompasses a timestamp value added by the issuing 3TCAC and marking the instant of its generation. As all the 3TCACs are synchronized, replayed messages are easily detected by the receiving 3TCAC which has to verify its current timestamp value in order to decide whether the message is replayed. All replayed messages will be simply ignored. Malicious RSs may also try to usurp the identity of other RSs or modify the integrity of the signalling message in order to tamper with the results of QoS estimation. This attack is detected since all signalling messages are created by the 3TCACs which also sign them. Note that the 3TCACs store the public keys of their subordinates and their superordinate in order to verify the signatures of the exchanged messages.

The DoS attacks may be caused by malicious RSs which refuse to forward the signalling messages carrying the QoS information or participating in evaluating this information. In order to tolerate the attacks targeting the QoS estimation, each 3TCAC is configured to raise an exception if the periodic signalling message is not received within a pre-configured threshold time period. In this case, the 3TCAC increments its own *probably-malicious-RS* variable and the *probably-malicious-RS* variable of the considered subordinate and an alert will bep sent to the managing MR-BS if the *probably-malicious-RS* variable reaches a threshold value.

Malicious RSs may try to pretend higher delay and jitter values through not forwarding the signalling messages on time. The receiving 3TCAC is strengthened with pre-configured threshold variables stating the maximum delay and the maximum jitter that characterize the considered link. When the evaluated delay (or jitter) exceeds the pre-configured threshold, the 3TCAC increments its own *probably* − *malicious* − *RS* variable and the *probably* − *malicious* − *RS*variable of the considered subordinate. When that variable reaches a pre-configured threshold, an alert is sent to the managing MR-BS. The MR-BS tolerates the malicious behavior through compensating the excess in delay (or jitter) by accelerating the transmission of the affected flows on other chosen branches on the route between it and the probably malicious RS. If needed, the MR-BS may coordinate with the other intermediate access entities (i.e., RSs and MR-BSs) on the route to the destination in order to compensate the malicious disturbance. The MR-BS also sends an alert to the operator's control center in order to request a maintenance procedure. Besides, RSs which negatively respond to a request of flows forwarding or to a compensation procedure are marked. When the history of negative responses of a certain RS reaches a pre-configured value, that RS will be considered as suspicious and an alert is sent to the MR-BS.

It is valuable to note that our 3TCAC architecture does not isolate the probably malicious RSs, especially when they are advertising higher QoS values. Contrarily, we propose to continue using them for forwarding traffic while compensating the disturbance caused by the malicious behavior using other legitimate RSs. The QoS compensation is done identically to the compensation in the group mobility case described earlier (i.e., the RS's handover). Besides, the MR-BS requests the maintenance procedure from the operator's control center. If the malicious RSs refuses to participate in the QoS estimation procedures, the QoS values that were validated in the previous period will be adopted until the maintenance is performed.

The proposed 3TCA architecture achieves a good level of intrusion tolerance since it rapidly detects and tolerates a large range of DoS and timely-based attacks. In particular, 3TCA may detect and tolerate Distributed DoS attacks (i.e., DDoS) that are mainly caused by colluding malicious SSs or colluding malicious RSs. Thanks to the configured threshold values and the synchronous behavior, a 3TCAC component is able to detect any DoS attack bombarding it or any DoS issued by malicious RSs refusing to correctly participate in the QoS estimation or in the data transfer. Moreover, the synchronous behavior of a 3TCA component enables it to easily detect replay attacks and time-based attacks aiming at tampering with the offered QoS. Attacks relative to the integrity of the transmitted messages and the authenticity of the participating entities in the route establishment and data transfer are easily detected and countered thanks to the authentication service guaranteed by the 3TCACs.

Besides, the 3TCA architecture induces a reduced number of false positives since the 3TCACs do not immediately declare an RS as malicious. They rather use two stages of variables and thresholds. If the first threshold is reached, the variable describing the malicious behavior of the RS is incremented until a second threshold is reached. More precisely, an RS is not declared malicious before the second threshold is reached. Therefore, we indirectly take into consideration the non-malicious causes of disturbance such as the saturation of a processor or the lateness or loss of data and signalling packets due to unfavorable transmission conditions. Meanwhile, the 3TCA architecture induces a reduced number of false negatives since it relies on the trusted 3TCACs. As described earlier, 3TCACs provide trusted security and time-related services which enable a rigorous control of the behavior of the malicious entities that are participating in the QoS estimation, the routes establishment and the data transfer.

The 3TCACs implement a trusted authentication service that secures the communication between a 3TCAE and its managing 3TCAC. In more detail, the trusted authentication service enables the 3TCAE to share a secret key with its 3TCAC. That key is then used to establish a secure channel between the 3TCAE and the correspondent 3TCAC over which the exchanged data may be encrypted. The trusted authentication service also enables each 3TCAE to authenticate its managing 3TCAC and ensures that the compromise of old keys does not enable a passive attacker to compromise future keys and does not allow an active adversary to fulfill impersonation. The trusted authentication service provided by 3TCACs builds on the Local Authentication service offered by the TTCB simply because we wish to provide the same protection level assessed and verified in [19]. Besides, the 3TCACs of our proposed architecture use asymmetric cryptography in order to verify the integrity of the messages they exchange between them and authenticate the issuers.

The 3TCACs use the trusted duration measurement service, the trusted absolute timestamping service, the trusted authentication service and the trusted random number generation service of the TTCB components in order to evaluate the handover delays in a trusted manner and to secure the communication between the 3TCAEs and the 3TCACs. Nevertheless, the 3TCACs implement the trusted intrusion-related service which is in charge of inspecting particular variables that are used to detect intrusions if their values exceed pre-defined thresholds. Therefore, 3TCACs may detect specific implementations of DoS, DDoS, and replay attacks and may identify the suspicious RSs and tolerate their malicious behaviors. Moreover, the 3TCACs are used to evaluate the handover duration and then compensate the induced delays in a trusted manner through processing CAC on chosen links. In a similar way, the negative impact of detected intrusions on the agreed QoS is evaluated and then compensated.

6 Performance evaluation

In this section, we evaluate the performance of the 3TCA architecture and determine whether the obtained results may be generalized.

6.1 Mathematical representations

Let SS_j and D_j be two communicating SSs belonging to the MRS network. Let SS_j be managed by a mobile RS_m and D_j managed by RS_{D_j}. Let MR-BS(RS_m) be the MR-BS managing RS_m and MR-BS(RS_{D_j}) be the MR-BS managing RS_{D_j}. We define HO_d as the delay required by RS_m to perform the handoff process. This delay includes the delay required by the managed SSs to re-estimate the QoS available on the edge links between them and the mobile RS_m in its new position. Let $Ad_{i,j}$ be the agreed delay for flow i of the managed SS_j. Therefore, $Ad_{i,j} = \text{Min}[\,Ad1_{i,j}, Ad2_{i,j}]$, where $Ad1_{i,j}$ is the agreed delay for flow i of the managed SS j before handoff and $Ad2_{i,j}$ is the agreed delay for the flow i of the managed SS j after the handoff and the QoS re-estimation. Let R be the set of the available routes between MR-BS(RS_m) and RS_{D_j}. The following cases are considered.

- If $HO_d < Ad_{i,j}$, MR-BS(RS_m) needs to find $r \in R$ that fulfills the updated delay value $Ad_{i,j} = Ad_{i,j} - HO_d$ based on the QoS information exchanged with the other MR-BSs and collected on the backbone. After that, the MR-BS(RS_m) will perform a CAC on the chosen route. Let $del_{i,j}$ be the delay required by MR-BS(RS_m) to perform the CAC on the selected route r. This procedure is the first delay compensation attempt.
- If $HO_d < Ad_{i,j}$ and $\nexists r$ fulfilling $Ad_{i,j} = Ad_{i,j} - HO_d$, we perform a second delay compensation attempt while adopting either a linear approach or an exponential approach.
  - If the linear approach is adopted, choose $y \in \aleph$ and $n \in \aleph$ and then launch n parallel CAC procedures on n routes in order to satisfy the condition $(Ad_{i,j} - (HO_d + del_{i,j} - ky)) > 0;\ k \in [\,1, n]$. Then select the route that satisfies $\text{Min}[\,Ad_{i,j} - (HO_d + del_{i,j} - ky)]$ as the route on which the flow will be transmitted.
  - If the exponential approach is adopted, choose $y \in \aleph$ and $n \in \aleph$, where $n = 2^k$ and then launch $k+1$ parallel CAC procedures on $k+1$ routes in order to satisfy the condition $(Ad_{i,j} - (HO_d + del_{i,j} - 2^h y)) > 0;\ h \in [\,0, k]$. Then select the route that satisfies $\text{Min}[\,Ad_{i,j} - (HO_d + del_{i,j} - 2^h y)]$ as the route on which the flow will be transmitted.
- If $HO_d > Ad_{i,j}$, we perform a delay compensation attempt while adopting either a linear approach or an exponential approach.
  - If the linear approach is adopted, choose $y \in \aleph$ and $n \in \aleph$ and then launch n parallel CAC procedures on n routes in order to satisfy the condition $(Ad_{i,j} - (HO_d - ky)) > 0;\ k \in [\,1, n]$. Then select the route that satisfies $\text{Min}[\,Ad_{i,j} - (HO_d - ky)]$ as the route on which the flow will be transmitted.
  - If the exponential approach is adopted, choose $y \in \aleph$ and $n \in \aleph$ where $n = 2^k$ and then launch $k+1$ parallel CAC procedures on

$k+1$ routes in order to satisfy the condition $(\mathrm{Ad}_{i,j} - (\mathrm{HO}_d - 2^h y)) > 0$; $h \in [0, k]$. Then select the route that satisfies $\mathrm{Min}[\mathrm{Ad}_{i,j} - (\mathrm{HO}_d - 2^h y)]$ as the route on which the flow will be transmitted.

Note that the HO_d may be replaced by $\mathrm{Intrusion}_d$ which is the delay induced by an intrusion in case of QoS compensation under intrusions.

Now, let nbr_{ss} be the number of SSs managed by the mobile RS RS_m. The HO_d value (i.e., that encompasses the delays of handoff and QoS re-estimation) according to the number of managed SSs can be given by the formula:

$$\mathrm{HO}_d = \alpha + \exp(nbr_{SS}), \tag{1}$$

where α is the mean delay of a hard IEEE 802.16j handoff.

Let *nbrHopsBS* be the number of intermediate hops from the RS_m til MR-BS(RS_m). Let $nbrHopsBSDest_r$ be the number of hops from MR-BS(RS_m) til RS_{D_j} on the route r where $r \in R$. We define the delay of transmitting the DSA-REQ or DSC-REQ message from the RS_m to MR-BS(RS_m) as $d_{\text{REQ-BS}}$. Meanwhile, we define the delay of transmitting the corresponding DSA-RSP or DSC-RSP from the MR-BS(RS_m) til the RS_m as $d_{\text{RSP-BS}}$. Besides, we define the delay of transmitting the DSA-REQ or DSC-REQ message on the intermediate nodes from MR-BS(RS_m) to the destination RS_{D_j} on a particular route r as $(d_{\text{REQ -inter}})_r$ and we define $(d_{\text{RSP-inter}})_r$ as the delay of transmitting the corresponding DSA-RSP or DSC-RSP message from RS_{D_j} to MR-BS(RS_m) on the intermediate nodes of a particular route r; $r \in R$. These delays on a particular route r are given by the following formulae:

$$d_{\text{REQ-B}} = \sum_{l=0}^{nbrHopsBS} (\mathrm{dtrans}_{\mathrm{REQ}})_l \tag{2}$$

$$d_{\text{RSP-BS}} = \sum_{l=0}^{nbrHopsBS} (\mathrm{dtrans}_{\mathrm{RSP}})_l \tag{3}$$

$$(d_{\text{REQ-inter}})_r = \sum_{l=0}^{nbrHopsBSDest_r} (\mathrm{dtrans}_{\mathrm{REQ}})_l, \tag{4}$$

where $(\mathrm{dtrans}_{\mathrm{REQ}})_l$ is the delay required for transmitting a DSA-REQ or DSC-REQ message on the wireless link l and $(\mathrm{dtrans}_{\mathrm{RSP}})_l$ is the delay required for transmitting a DSA-RSP or a DSC-RSP message on the wireless link l.

As MR-BS(RS_m) knows the periodically updated values of the QoS offered by the routes belonging to R, it may select the most appropriate route $r_{\text{selected}} \in R$ that offers the required QoS after handover. We define the delay $\mathrm{del}_{i,j}$ required to process a first delay compensation as the delay of transmitting the DSA-REQs or DSC-REQs on the selected route r_{selected} then receiving the corresponding DSA-RSP or DSC-RSPs on the same route:

$$\mathrm{del}_{i,j} = d_{\text{REQ-BS}} + (d_{\text{REQ-inter}})_{r_{\text{selected}}} + (d_{\text{RSP-inter}})_{r_{\text{selected}}} + d_{\text{RSP-BS}} \tag{5}$$

If $\mathrm{HO}_d < \mathrm{Ad}_{i,j}$ but the first delay compensation attempt is fruitless (i.e., for example due to an update of the QoS values offered by the routes), a second delay compensation will be performed. Let $(\mathrm{del2lin}_{i,j})_n$ be the delay required to perform the second delay compensation on n selected routes with updated QoS values while adopting the linear approach. Moreover, let $R = \{r_{\text{selected}}\} \smile (R2_{\text{sel}})_n \smile (\overline{R2_{\text{sel}}})_n$ where $(R2_{\text{sel}})_n$ is the set of the most appropriate n routes on which a new CAC procedure will be processed and $(\overline{R2_{\text{sel}}})_n$ is the set of the remaining routes. $(\mathrm{del2lin}_{i,j})_n$ is given by the following formula:

$$(\mathrm{del2lin}_{i,j})_n = \mathrm{del}_{i,j} + n * d_{\text{REQ-BS}} + d_{\text{RSP-BS}} + \sum_{r \in (R2_{\text{sel}})_n} \left[(d_{\text{REQ-inter}})_r + (d_{\text{RSP-inter}})_r \right], \tag{6}$$

note that, after processing the CAC on the n selected routes, the MR-BS(RS_m) will send a unique DSA-RSP or a DSC-RSP indicating the route offering the minimum delay on which the flow will be transmitted.

Let $(\mathrm{del2exp}_{i,j})_k$ be the delay required to perform the second delay compensation while adopting the exponential approach. Let us remember that $n = 2^k$. In this case, a new CAC procedure is processed on the most appropriate $k+1$ routes. Meanwhile, let $R = \{r_{\text{selected}}\} \smile (R2_{\text{sel}})_k \smile (\overline{R2_{\text{sel}}})_k$ where $(R2_{\text{sel}})_k$ is the set of the most appropriate $k+1$ routes on which the CAC procedure will be processed and $(\overline{R2_{\text{sel}}})_k$ is the set of the remaining routes. $(\mathrm{del2exp}_{i,j})_k$ is given by the following formula:

$$(\mathrm{del2exp}_{i,j})_k = \mathrm{del}_{i,j} + (k+1) * d_{\text{REQ - BS}} + d_{\text{RSP-BS}} + \sum_{r \in (R2_{\text{sel}})_k} \left[(d_{\text{REQ-inter}})_r + (d_{\text{RSP-inter}})_r \right], \tag{7}$$

note that, after processing the CAC on the $k+1$ selected routes, the MR-BS(RS_m) will send a unique DSA-RSP or a DSC-RSP indicating the route offering the minimum delay on which the flow will be transmitted.

If $\mathrm{HO}_d > \mathrm{Ad}_{i,j}$ and we adopt the linear approach to compensate the handover delay, $(\mathrm{dellin}_{i,j})_n$ will be the delay of such a compensation. In this case, the CAC procedure is straightforward processed on the most appropriate n routes that may fulfill the QoS requirements. Let $R = (R1_{\text{sel}})_n \smile (\overline{R1_{\text{sel}}})_n$ where $(R1_{\text{sel}})_n$ is the set of the most appropriate n routes on which the CAC procedure will be

processed and $(\overline{R1_{\text{sel}}})_n$ is the set of the remaining routes. $(\text{dellin}_{i,j})_n$ is given by the following formula:

$$(\text{dellin}_{i,j})_n = n * d_{\text{REQ-BS}} + d_{\text{RSP-BS}} + \sum_{r \in (R1_{\text{sel}})_n} [\,(d_{\text{REQ-inter}})_r + (d_{\text{RSP-inter}})_r\,], \tag{8}$$

note that, after processing the CAC on the n selected routes, the MR-BS(RS_m) will send a unique DSA-RSP or a DSC-RSP indicating the route offering the minimum delay on which the flow will be transmitted.

Lastly, if $\text{HO}_d > \text{Ad}_{i,j}$ and we straightway adopt the exponential approach to compensate the handover delay, a CAC procedure will be processed on the most appropriate $k+1$ routes. Let us remember that $n = 2^k$. Let $R = (R1_{\text{sel}})_k \smile (\overline{R1_{\text{sel}}})_k$ where $(R1_{\text{sel}})_k$ is the set of the most appropriate $k+1$ routes on which the CAC procedure will be processed and $(\overline{R1_{\text{sel}}})_k$ is the set of the remaining routes. $(\text{delexp}_{i,j})_k$ will be the delay of such compensation, it is given by the following formula:

$$(\text{delexp}_{i,j})_k = (k+1) * d_{\text{REQ-BS}} + d_{\text{RSP-BS}} + \sum_{r \in (R1_{\text{sel}})_k} [\,(d_{\text{REQ-inter}})_r + (d_{\text{RSP-inter}})_r\,], \tag{9}$$

note that, after processing the CAC on the $k+1$ selected routes, the MR-BS(RS_m) will send a unique DSA-RSP or a DSC-RSP indicating the route offering the minimum delay on which the flow will be transmitted.

Note that when $(R2_{sel})_n = (R1_{sel})_n$, $(\text{del2lin}_{i,j})_n$ will be obviously superior than $(\text{dellin}_{i,j})_n$ by $\text{del}_{i,j}$. Similarly, when $(R2_{sel})_k = (R1_{sel})_k$, $(\text{del2exp}_{i,j})_k$ will be obviously superior than $(\text{delexp}_{i,j})_k$ by $\text{del}_{i,j}$.

6.2 Simulation model

In order to estimate the performance of the 3TCA architecture for QoS routing and intrusion tolerance provision within the multihop relay network, we adopt the network shown in Figure 5. The network encompasses mobile N-LoS RSs such as the RS12, fixed LoS RSs such as RS11 and RS13 and a set of MR-BSs interconnected with a wireless backbone. The mobile RS12 hands over and enters into the coverage area of RS11. RS12 manages two SSs which are SS11 and SS12. Each SS has a unique flow to be transmitted. The fixed and mobile RSs along with the MR-BSs host trusted 3TCACs.

We propose to perform four simulations. To that aim, we wrote the code implementing the 3TCA features and then we fixed simulation scenarios and we computed the execution cost of the scenarios implementation. The first considered scenario consists of a mobile RS that moves with its two managed SSs. The other scenarios consider legitimate and malicious RSs performing QoS re-estimation. First, we only detect the attack, and then we detect the attack and we tolerate it through processing QoS compensation. This will be further detailed when we describe each simulation context.

The goal of the first simulation is to evaluate the connection establishment delay when QoS compensation is processed after handover. The second simulation intends to determine the overhead induced by the QoS compensation process after handover. In both simulations, we consider the scenario when RS12 hands over to enter into the coverage area of RS11. In the third simulation, we propose to estimate the processing delay when our network is under attack and we evaluate the cost of guaranteeing intrusion-tolerance and of performing QoS compensation after being the victim of an intrusion. Finally, the last simulation aims at determining the overhead caused by the QoS compensation in case of intrusion. Note that for the described simulations, we assume that the delay is proportional to the number of executed operations.

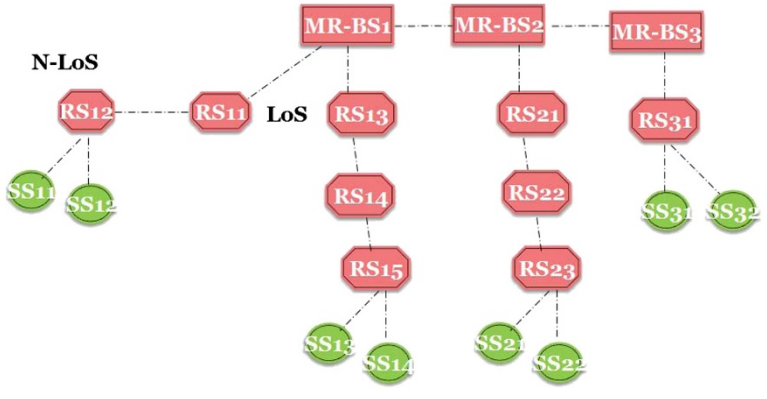

Figure 5 The chosen network for simulations.

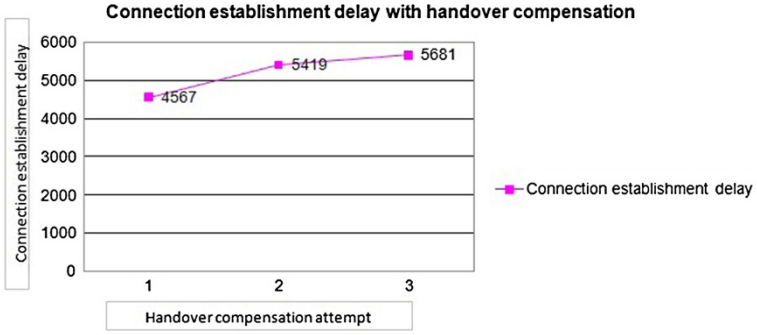

Figure 6 Number of operations executed when handover compensation is provided.

6.3 Evaluating the performance of the 3TCA's protocols

The first curve shown in Figure 6 estimates the number of operations executed when the RS12 moves with its managed SSs. The first attempt reflects the situation when the handover delay is smaller than the already agreed delay and when the handover compensation is successful from the first time. Therefore, the delay required to process the first attempt includes the delay described by Equation (6) along with the delay required to process the QoS re-estimation between the managed SSs and the handing over RS after handover. The second attempt reflects the situation when the handover delay is superior to the already agreed delay so that a linear approach is adopted. Therefore, the second attempt includes the delay described by the Equation (9) and the delay of QoS re-estimation. The third attempt reflects the situation when the handover delay is smaller than the already agreed delay and when the handover compensation is not successful from the first time so that a linear approach is adopted. Therefore, the third attempt includes the delay described by the Equation (7) along with the delay of QoS re-estimation. The simulated case assumes that for the two times where the linear approach has been adopted, $(R2_{\text{sel}})_n = (R1_{\text{sel}})_n$. Note that three points of the abscissa axis reflect the three main scenarios of the handover's impact compensation.

The experimental results confirm the theoretical results. More precisely, when the handover delay is smaller than the already agreed delay and the first compensation attempt is successful from the first time (i.e., the first point in the abscissa axis), only a CAC on a chosen route is performed and that route is able to provide the updated values of QoS. The delay of performing that successful first CAC is $\text{del}_{i,j}$. Moreover, when the handover delay is larger than the already agreed delay (i.e., the second point in the abscissa axis), only a first compensation attempt is performed. In this case, the CAC is processed on n routes and the resulting delay is $(\text{dellin}_{i,j})_n$. Note that $(\text{dellin}_{i,j})_n$ is superior than $\text{del}_{i,j}$ because we perform the CAC on n routes and not only on one chosen route. However, when the handover delay is smaller than the already agreed delay and the first compensation attempt is fruitless, we perform a second compensation attempt (i.e., the third point in the abscissa axis). As we simulate the case where $(R2_{\text{sel}})_n = (R1_{\text{sel}})_n$, $(\text{del2lin}_{i,j})_n$ will be superior than $(\text{dellin}_{i,j})_n$ because it includes $\text{del}_{i,j}$.

We may conclude that adopting the compensation induces some delay when it is not successful from the first

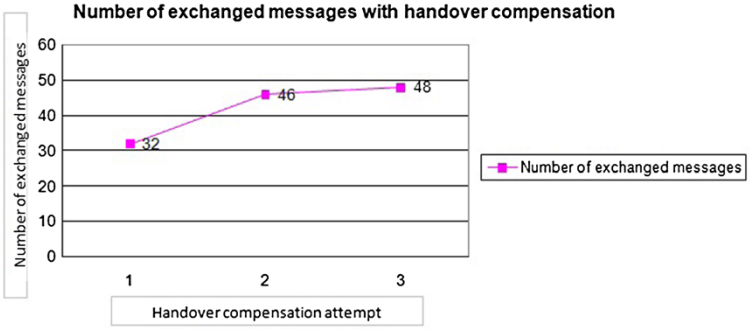

Figure 7 Number of messages exchanged when handover compensation is provided.

time. Note that the delay caused by an unsuccessful compensation for the first time is taken into consideration when trying the compensation for the second time.

The second curve shown in Figure 7 estimates the number of the signalling messages exchanged when the RS12 hands over with its managed SSs. The same simulation scenario of the estimation of the connection establishment delay with handover compensation is adopted. Note that the number of exchanged messages increases especially when the compensation is not successful from the first time, which is an expected behavior.

In order to evaluate the performance of the 3TCA architecture with respect to intrusion tolerance, we consider various situations in which some RSs become malicious during the periodic QoS estimation on the backbone. More precisely, we increase the number of malicious nodes and we observe the delay and overhead induced by detecting and tolerating the intrusions. The first point of the abscissa axis shown by Figures 8 and 9 reflects the situation when the QoS re-estimation procedure is normally processed (i.e., without facing any intrusions). The second point of the abscissa axis reflects the situation when the QoS re-estimation faces an intrusion at the RS23 level but the threshold value of the *probably-malicious-RS* variable is not yet reached. In this case, only the value of that variable is incremented without affecting the normal QoS re-estimation procedure. The third point of the abscissa axis reflects the situation when the QoS re-estimation faces an intrusion at both RS23 and RS31 levels. In this case, the intrusions are detected but the 3TCACs do not react because the *probably-malicious-RS* threshold is not yet reached. Lastly, the fourth point of the abscissa axis reflects the situation when RS23, RS31, and RS15 are malicious and a compensation procedure is processed for RS15 while the intrusions of RS23 and RS31 are only detected. The 3TCACs do not react because the *probably-malicious-RS* threshold is not yet reached for both RS23 and RS31. However, the delay estimated by RS14 is superior to the threshold and the threshold value of the *probably-malicious-RS* variable is reached. In that case, RS14 sends the *probably-malicious-RS-Alert* to the MR-BS1 which reacts by attempting an intrusion compensation on the link RS13 $\rightarrow$ MR-BS1. Note that the four points of the abscissa axis are particularly chosen to reflect the main intrusion tolerance behavior in case of attacks.

As shown by Figure 8, the processing delay increases in case of intrusion. Moreover, the more the number of malicious RSs increases, the more the intrusion tolerance processing delay augments. The delay of the compensation attempt is important as the MR-BS needs to ask the involved RSs of the compensation links in order to decide whether the compensation is possible.

Let us now further evaluate the performance of our proposed architecture regarding intrusion-tolerance. To achieve this, we propose to compare the simulation results obtained within the MMR networks context to those of an intrusion tolerant routing protocol within the mesh context. Such a routing protocol, called MERQIT, has been presented in [9]. We think that such a comparison is appropriate for two reasons. On one hand, both protocols address the combination of the intrusion tolerance property and the QoS provision within the context of mobile wireless networks. On the other hand, the MERQIT protocol relies on clusterheads in providing connection to Non Line of Sight mobile subscribers. Indeed, the clusterheads considered by MERQIT may be compared to the relays considered by the MMR networks; the main difference between them is that the relays are managed by the network operator while the clusterheads are not.

We note that the simulation results obtained for the 3TCA architecture in case of intrusion confirm the results describing the MERQIT protocol behavior in case of intrusions. Particularly, Figure 8 shows the processing delay according to the number of malicious nodes within a mesh network. We notice that the processing delay does not highly increase when a malicious clusterhead is only detected without being isolated, but that the same delay becomes very important when the number of malicious

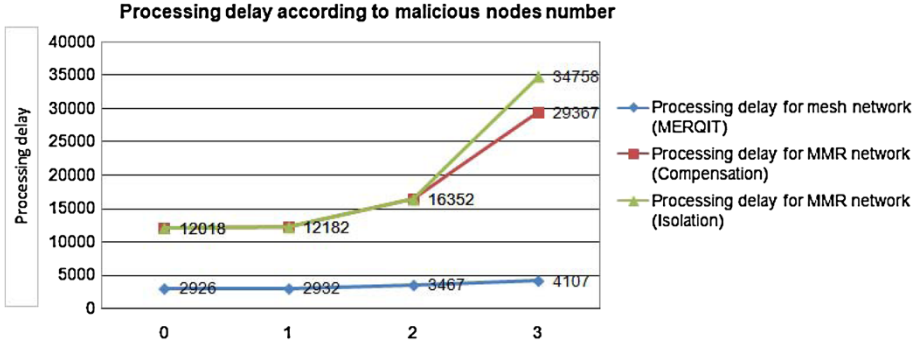

Figure 8 Processing delay according to malicious nodes number.

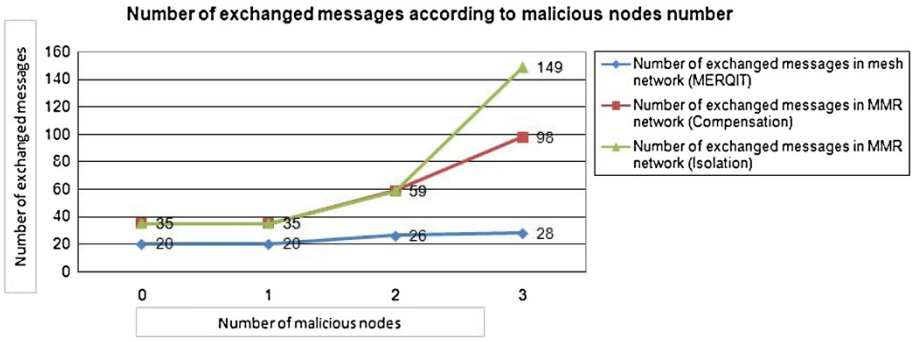

Figure 9 Number of exchanged messages with respect to malicious nodes number.

access nodes increases (i.e., for example when a router presents malicious entities and should broadcast its status to its neighbors or when a clusterhead should be completely isolated) [9].

Nevertheless, we notice that MERQIT achieves better delays in the case of intrusions. First, because the simulated case of MERQIT is limited to QoS provision in terms of delays and does not consider QoS provision in terms of jitter, and second, because the MERQIT protocol isolates the malicious clusterheads or routers and asks the neighboring access nodes to take over if the QoS can be met. For example, since the same MN is managed by two clusterheads and if the managing router discovers that the first clusterhead is malicious, it will ask the second clusterhead to take over. In this case, minor treatments are processed as the second clusterhead has already all the information required to fulfill the failover. The 3TCA architecture rather processes QoS compensation by negotiating updated QoS values with candidate RSs of chosen links in order to accelerate the flows affected by the intrusion. Figure 8 also shows that adopting the isolation principle of the MERQIT protocol within the MMR networks is not appropriate. More precisely, when the probably malicious RS15 and RS14 are isolated, the whole branch of the tree rooted at RS14 becomes isolated. Consequently, the managed SS13 and SS14 are isolated and need to be rescued. In order to minimize the isolation impact, MR-BS1 needs to ask the neighboring MR-BS2 whether it is managing an RS that is in LoS with the isolated SSs and whether that candidate rescuing RS belongs to a path that fulfills the SSs' QoS requirements. If it is the case, the isolated SSs will handover in order to attach to the rescuing RS and then they will process an edge QoS estimation with their new managing access node. This costly procedure in terms of processing delay may fail especially when the number of isolated SSs is important. For these reasons, we argue that adopting compensation in case of intrusion is more suited to the MMR context.

Now let us estimate the overhead when tolerating intrusions for our proposed scheme relative to MMR networks and let us adopt the same simulation scenario considering the processing delay with intrusion tolerance. As shown in Figure 9, the number of exchanged messages is the same when the normal QoS re-estimation is processed and when the intrusion affecting one RS is detected but not tolerated. That number increases when the number of malicious RSs increases and when the MR-BS1 tries to compensate the intrusion.

The overhead estimation results obtained in the MMR context confirm those obtained for MERQIT. In fact, we notice that the number of exchanged messages does not highly vary when a clusterhead is detected as malicious without being isolated. However, the overhead increases when the number of malicious access nodes increases. For example, the overhead augments when a router on the backbone should broadcast its status or when a manager should be isolated, [9]. Note that the number of exchanged messages in the context of MMR networks is superior to that number in the context of mesh networks mainly because we have implemented the jitter estimation within MMR networks. In fact, as stated in Section 2.3, the jitter estimation over a wireless link requires the exchange of at least 9 packets; thus, increasing the overhead of both QoS estimation and QoS compensation. Note that adopting isolation instead of compensation in case of intrusion within MMR networks induces an important overhead that increases exponentially with the number of managed SSs. This overhead results from the CAC processed by each neighboring MR-BS and its managed RSs along with the overhead caused by processing the handover operation and implementing the edge QoS re-estimation with the rescued SSs.

The simulations adopted in this article estimate the complexity of the 3TCA architecture in terms of processing delay and induced overhead. We may conclude that the obtained results are expected since one observes

an increase of complexity and overhead when compensation is processed. This increase is the normal cost of an advanced management of QoS that adopts several attempts of compensation in case of handover and attacks in order to preserve the initial level of QoS or come close to that level. The simulation results obtained when adopting the pre-described simulation model may be generalized since we simulated the main scenarios of handover's impact compensation and of intrusion tolerance behaviors.

7 Conclusion

In this article, we proposed a novel 3TCA architecture for MMR networks. Our architecture is based on trusted 3TCACs and it addresses the RSs' mobility through compensating the handover-induced delays. Besides, intrusions are detected and their side effects are compensated in the multihop relay context in order to continue providing the agreed QoS level despite intrusions. Simulations show an increase in the complexity reflecting the cost for intrusion tolerance guarantee and advanced mobility management.

Competing interests
Both authors declare that they have no competing interests.

Received: 27 February 2012 Accepted: 8 February 2013
Published: 8 April 2013

References

1. IEEE standard for local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems, Amendment 1: Multihop Relay Specification, IEEE std 802.16j-2009, IEEE Computer Society and IEEE Microwave Theory and Techniques Society Std. (2009)
2. IEEE standard for local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems, IEEE std 802.16-2009, IEEE Computer Society and IEEE Microwave Theory and Techniques Society Std (2009)
3. D Powell, R Stroud, Conceptual Model and Architecture of MAFTIA . Tech. Rep. MAFTIA deliverable D21 (2003). http://webhost.laas.fr/TSF/cabernet/maftia/deliverables/D21.pdf
4. J Chang, W Hsiao, J Chen, H Chao, in *International Conference on Ubiquitous Information Technologies & Applications (ICUT)*. Mobile relay stations navigation-based self-optimization handover mechanism in WiMAX networks (Fukuoka, 2009), pp. 1–5, ISBN: 978-1-4244-5131-9
5. B Chang, S Hsieh, Y Liang, D Wang, in *International Conference on Advanced Information Technologies (AIT)*. QoS guarantee-based fast handoff in IEEE 802.16j WiMAX MMR networks (Taiwan, 2009)
6. D-H Kim, S-S Kim, Y-H Lee, Speed-sensitive handover scheme over IEEE 802.16 multi-relay networks. J. Inf. Process. Syst. **6**(3), 403-412 (2010)
7. S Ann, KG Lee, HS Kim, in *International Conference on Sensor Technologies and Applications*. A path selection method in IEEE 802.16j mobile multi-hop relay networks (Cap Esterel, 2008), pp. 808 - 812. ISBN: 978-0-7695-3330-8
8. N Krichene, N Boudriga, in *IFIP Book Series, Ad-Hoc Networking*, vol. 212. On a QoS intrusion tolerant routing protocol in ad-hoc networks (Santiago, Chile, 2006), pp. 29 - 46
9. N Krichene, N Boudriga, in *IFIP Int. Conference on Wireless and Optical Communications Networks (WOCN)*. Intrusion tolerant routing for mesh networks (Singapore, 2007), pp. 1–7. ISBN: 1-4244-1005-3
10. V Genc, S Murphy, Y Yu, J Murphy, IEEE 802.16j relay-based wireless access networks: an overview. IEEE Wirel. Commun. **15**, 56–63 (2008)
11. WG Bouricius, WC Carter, PR Schneider, in *Association for Computing Machinery (ACM)*. Reliability modeling techniques for self-repairing computer systems (New York, 1969), pp. 295–309
12. The MAFTIA project Internet site, http://webhost.laas.fr/TSF/cabernet/maftia/programme/
13. R Ma, L Xing, H Michel, in *IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC)*. Fault-intrusion tolerant techniques in wireless sensor networks (Indianapolis, 2006), pp. 85-94
14. F Tang, M Guo, M Li, C Wang, M Dong, Secure routing for wireless mesh sensor networks in pervasive environments. Int. J. Intell. Control Syst. **12**(4), 293–306 (2007)
15. A Perrig, R Szewczyk, JD Tygar, V Wen, DE Culler, SPINS: security protocols for sensor networks. J. Wirel. Netw. **8**(5), 521–534 (2002)
16. N Ferreira, P Verrissimo, Complete specification of APIs and protocols for the MAFTIA middleware. Tech. Rep. MAFTIA delivrable D9 (2002). http://webhost.laas.fr/TSF/cabernet/maftia/deliverables/D9.pdf
17. V Toncar, Aboutjitter Voip basics. http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Jitter.html
18. AS Khan, N Fisal, S Kamilah, M Abbas, Efficient distributed authentication key scheme for multi-hop relay in IEEE 802.16j network. Int. J. Eng. Sci. Technol. **2**(6), 2192–2199 (2010)
19. A Adelsbach, S Creese, Final Report on Verification and Assessment. Tech. Rep. MAFTIA delivrable D22 (2003). http://webhost.laas.fr/TSF/cabernet/maftia/deliverables/D22.pdf

doi:10.1186/1687-1499-2013-100
Cite this article as: Krichene and Boudriga: **Intrusion tolerant QoS provision in mobile multihop relay networks.** *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:100.