

## Research Article

# Fault Tolerant Mechanism for Multimedia Flows in Wireless Ad Hoc Networks Based on Fast Switching Paths

Juan R. Diaz,<sup>1</sup> Jaime Lloret,<sup>1</sup> Jose M. Jimenez,<sup>1</sup>  
Sandra Sendra,<sup>1,2</sup> and Joel J. P. C. Rodrigues<sup>2</sup>

<sup>1</sup> Instituto de Investigación para la Gestión Integrada de Zonas Costeras (IGIC), Universidad Politécnica de Valencia, Camino Vera s/n, 46022 Valencia, Spain

<sup>2</sup> Instituto de Telecomunicações, Universidade da Beira Interior, Rua Marquês de Ávila e Bolama, 6201-001 Covilhã, Portugal

Correspondence should be addressed to Jaime Lloret; [jlloret@dcom.upv.es](mailto:jlloret@dcom.upv.es)

Received 14 July 2014; Accepted 21 July 2014; Published 2 September 2014

Academic Editor: Jingjing Zhou

Copyright © 2014 Juan R. Diaz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multimedia traffic can be forwarded through a wireless ad hoc network using the available resources of the nodes. Several models and protocols have been designed in order to organize and arrange the nodes to improve transmissions along the network. We use a cluster-based framework, called MWAHCA architecture, which optimizes multimedia transmissions over a wireless ad hoc network. It was proposed by us in a previous research work. This architecture is focused on decreasing quality of service (QoS) parameters like latency, jitter, and packet loss, but other network features were not developed, like load balance or fault tolerance. In this paper, we propose a new fault tolerance mechanism, using as a base the MWAHCA architecture, in order to recover any multimedia flow crossing the wireless ad hoc network when there is a node failure. The algorithm can run independently for each multimedia flow. The main objective is to keep the QoS parameters as low as possible. To achieve this goal, the convergence time must be controlled and reduced. This paper provides the designed protocol, the analytical model of the algorithm, and a software application developed to test its performance in a real laboratory.

## 1. Introduction

A wireless ad hoc network is a decentralized structure by nature. This type of network does not depend on a preexisting infrastructure such as wireless routers or access points in managed wireless networks. Instead of this, each node participates in routing tasks by forwarding data to other nodes. The purpose of these nodes is determined dynamically based on the network connectivity. An ad hoc network typically refers to any set of nodes where all of them have the same status within the network and are free to associate with any other node in the radio coverage [1]. An ad hoc network is composed of several nodes connected by radio links. These links are limited by the resources of each node such as its power consumption, transmission power, computing power, memory, performance properties, and radio link properties (e.g., the duration of the connection, signal loss, interference, and noise). As radio links can be connected or disconnected at any time, a functioning network must be able to cope

this dynamic restructuring, preferably in a timely, efficient, reliable, robust, and scalable way [2].

Ad hoc networks are suitable for a variety of applications such as emergency situations, armed conflicts, natural disasters [3], or environmental monitoring [4], where nodes are not dependent on a central node. This feature can improve the scalability of these networks compared to managed wireless networks. Furthermore, we can find lots of hardware available for their implementation [5]. Among lots of applications that can be implemented by the ad hoc networks, we can find the multimedia content distribution. Design of a multimedia ad hoc network for a certain application is influenced by several factors, such as fault tolerance [6], scalability, production costs, operating environment, network topology, hardware constraints, transmission media, and power consumption [7]. Moreover, there exist additional factors which affect the efficiency of multimedia communication in ad hoc networks such as high bandwidth demand, multimedia coding techniques, application-specific quality of service (QoS)

requirements, and delay bounds, among others. These factors are some of the most important ones as they serve as guidelines to design communication protocols, multimedia applications, and algorithms for efficient multimedia communications in ad hoc networks.

Routing protocols for multimedia ad hoc and wireless sensor networks can be divided into location-based, hierarchical, and flat networks [8]. All of them try to solve and improve their problems and requirements. On the one hand, a hierarchical architecture seems to be more beneficial than flat architectures in terms of less energy consumption, higher functionality, better scalability, and reliability. When multimedia ad hoc networks or wireless sensor networks use a hierarchical architecture, such as I cluster-based networks, we can see that nodes have different roles or tasks [9]. This reduces the energy consumption. In addition, the Cluster-Head (CH) can perform data aggregation which avoids unnecessary data transmission and reduce the total consumed bandwidth. Finally, the non-CH can remain in standby mode (turning off the radio interface) after transmitting its packets, reducing energy consumption, and avoiding communication collisions and interferences.

As [10] shows, some of the main requirements of the hierarchical routing protocol in multimedia ad hoc and wireless sensor networks is to minimize the signaling overhead for cluster formation and guarantee multimedia retrieval and retransmission [11]. In addition, the routing protocol should be able to ensure good levels of QoS and Quality of Experience (QoE) in routes where multimedia and scalar data are transmitted. Thus, this protocol should provide scalability and reliability. However, most of current hierarchical routing protocols for multimedia ad hoc and wireless sensor networks do not take into account such key features.

Finally, it is important to select the appropriate nodes to have connections between failure nodes or when a radio link is lost. We must also consider that there are applications where the use of mixed networks with mobile and static nodes is considered. In these cases, the protocol should be able to implement handover mechanisms. Moreover, a protocol which implements a reliable handover mechanism should guarantee some parameters such as faded signal-to-noise ratio, residual channel capacity, and connection life time [12]. The goal of a good implementation should be a network with low handover and small end-to-end delays for multimedia applications. Packet loss ratios as well as efficient network selection processes will also be enhanced in comparison with other proposals which do not consider these parameters [13].

In this paper, we are going to propose a new fault tolerant mechanism for wireless ad hoc networks. The algorithm will be developed inside the MWAHCA framework following [10, 11] specifications. The main objective of our proposal is to have an algorithm to provide fault tolerance to each multimedia flow in a wireless ad hoc network running a protocol based in the MWAHCA architecture. We are not going to modify the original architecture but we are going to propose a protocol enhancement and extension. This new algorithm will be analyzed theoretically and experimentally in the laboratory.

The rest of this paper is structured as follows. Section 2 shows some previous proposals on multimedia ad hoc and sensor networks where the main features of these networks are analyzed. Section 3 explains the fault tolerance behavior included in the MWAHCA architecture, then, our proposal is explained, and the protocol extension in order to implement this mechanism is described. Section 4 exposes theoretical calculations to estimate the convergence time and QoS parameters for this proposal. Section 5 focus on the experimental study performed. Finally, conclusion and future work are presented in Section 6.

## 2. Related Work

Nowadays, we can find some works about routing protocols to solve the problems we can find in multimedia wireless ad hoc and sensor networks. Most of these proposals are tested in network simulators and very few are developed for real environments.

Researchers like Pagani and Rossi [14] presented a reliable broadcast protocol designed for mobile ad hoc networks. The proposed protocol was implemented on top of the wireless MAC protocol which sits over the clustering protocol. The protocol also provides tolerance in communication failures and host mobility. The reliable broadcast service ensures that all hosts in the network deliver the same set of messages to the upper layer. Their results show that the proposal provides high broadcast and multicast services as an efficient and reliable alternative to the flooding method.

Gupta and Younis proposed in [15] a high-energy gateway node that acts as a centralized manager to handle the sensors. The gateway serves as a hop to relay data and commands from sensors to a distant node. They have introduced a two phase in the communication process to apply fault tolerance approach in sensor networks. This system allows detecting and recovering sensors from the failed gateways without shutting down or recluster the system. Their approach enables fault tolerance in the system by performing periodic checks on the status of the gateways. Sensors managed by a faulty gateway are recovered by reassociating them to other clusters based on backup information created during the clustering time.

Tai et al. [16] developed an algorithm for detecting failures in distributed systems formed over ad hoc wireless networks. By exploiting the cluster-based communication architecture, the system is able to make the failure detection service scalable and resilient to link failures. Moreover, the proposed protocol allows intra- and intercluster communication to be resilient to message loss and node failure by taking advantage of the inherent message redundancy in ad hoc wireless networks. Kuhn et al. [17] also studied distributed approximation algorithms for fault tolerant clustering in wireless ad hoc and sensor networks.

Other authors such as Xue and Nahrstedt [18] proposed a new routing service named best-effort fault tolerant routing (BFTR). The design of BFTR is not to detect whether routing path consists of any misbehaving node, but to evaluate its routing feasibility based on its end-to-end connection. Analytical and experimental results demonstrate that BFTR

greatly improves the ad hoc routing performance in the presence of misbehaving nodes.

On the other hand, some researchers proposed fault tolerant routing protocols for ad hoc networks. Among them, Boukerche et al. [19] proposed two routing protocols: periodic, event-driven, and query-based protocol (PEQ) and its variation CPEQ. They are two fault tolerant and low-latency algorithms that meet sensor network requirements for critical conditions supervision in context-aware physical environments. PEQ uses a small amount of information for the routing mechanism (basically the hop level and routing table). When a failure is detected, unlike other solutions that use three way protocols, PEQ broadcasts a SEARCH packet to its neighbors and receives a reply with their hop level and identification. The neighbor with lower hop level is chosen as the new destination. In this way, loop back is avoided. Bheemarjuna Reddy et al. [20] presented MuSeQoR, a new multipath routing protocol that tackles the twin issues of reliability (protection against failures of multiple paths) and security while ensuring minimum data redundancy. Reliability is addressed on the context of both erasure and corruption channels. Authors also quantified the security of their protocol in terms of the number of eavesdropping nodes. The requirements of reliability and security in a session are specified by a user and are related to the parameters of the protocol adaptively. This relationship shows how the protocol attempts to simultaneously achieve reliability and security. In addition, the protocol minimizes the redundancy by using optimal coding schemes and by dispersing the original data.

Chao and Chang [21] proposed a fault tolerant routing protocol called Sensor On-demand Multi-path Distance Vector Reliable (SOMDV-R) routing protocol for wireless sensor networks which support reliable data delivery. This protocol takes into account reliability demand and link quality to determine the number of desired paths. Results show that, in comparison with Ad hoc On-demand Multi-path Distance Vector (AOMDV) and AODV, SOMDV-R can achieve higher packet delivery rate, lower routing overhead, and lower mean latency, upon different channel conditions. Melamed et al. [22] also presented a simple fault tolerant protocol called Octopus. This proposal showed great efficiency in routing tasks for large MANETs which can work with mobile nodes.

Finally, regarding power consumption, Antoniou et al. [23] proposed a new energy efficient and fault tolerant protocol for data propagation in wireless sensor networks. It is called Variable Transmission Range Protocol (VTRP). The main idea of this protocol is based on the varying range of data transmissions. The protocol exhibits high fault tolerance, by bypassing obstacles or failed sensors, and increases network lifetime (since critical sensors, that is, close to the control center, are not overused). The protocol was evaluated and compared, in terms of performance measures and energy consumption, with representative protocols such as Licklider Transmission Protocol (LTP). The results show that their protocol achieves significant improvements in energy efficiency and network lifetime.

As we have seen, most of these works are focused on solving some of these issues individually. Our proposal tries

to solve the problem of fault tolerance basing its operation on a smart protocol. In addition, this proposal has been tested in a real environment to demonstrate its performance.

### 3. Fault Tolerance in the Multimedia Wireless Ad Hoc Network

In this section, we are going to explain how fault tolerance is managed in the MWAHCA architecture. As proposed in [11], MWAHCA is a multimedia wireless ad hoc network architecture based on clusters. The aim of this model is to provide the reference framework in order to build an efficient and robust network protocol which can improve QoS and QoE parameters on multimedia transmissions through ad hoc wireless networks. The architecture has been successfully implemented in a wireless ad hoc protocol [13] and tested on IEEE 802.11 networks and wireless sensor networks (WSN) [10].

In order to achieve our goals, a new decision algorithm is proposed to achieve fault tolerance for multimedia wireless ad hoc networks. The proposed mechanism runs independently for each multimedia flow when there is a node failure in the network. In order to avoid the degradation of QoS parameters such as latency, jitter, or packet loss, the algorithm looks for a fast switching path while the system is being restored or a new optimal path is being recalculated and guaranteed. The fast switching path introduced by the algorithm is not a guaranteed route and it will be dismantled when an optimal path is found. The main objective of this proposal is to reduce the convergence time when a network failure happens and the end user could keep receiving multimedia packets as soon as possible after the network failure. As the MWAHCA architecture is being used as framework base, the algorithm must be consistent with this model and no architecture modifications are allowed.

Finally, an extension of the protocol implemented in [10] is proposed. This protocol is based on the MWAHCA model. We introduce the algorithm and the analytical model to test the consistency of the proposal. Several types of messages are added to the original table and some new exchange messages are defined.

#### 3.1. Adding Fault Tolerance to the MWAHCA Architecture.

In the MWAHCA architecture, nodes are grouped in several clusters following the Multimedia Init Profile (MIP) configuration as described in [11]. The MIP configuration collects the hardware characteristics that a node must fulfill in order to join a cluster. Following the reference model, only adjacencies between nodes with the same MIP are allowed; thus each cluster in the network is created by joining nodes with similar behavior and available resources. MIP configuration is initially assigned to cluster nodes. It limits the types of multimedia flows that can be managed by a specific cluster node. In this paper, we are going to focus only on the fault tolerance mechanism based on fast flow recovery, but we are not going to go deep inside the whole network structure or in the different process to make or destroy an adjacency. The starting point of this research is a multimedia

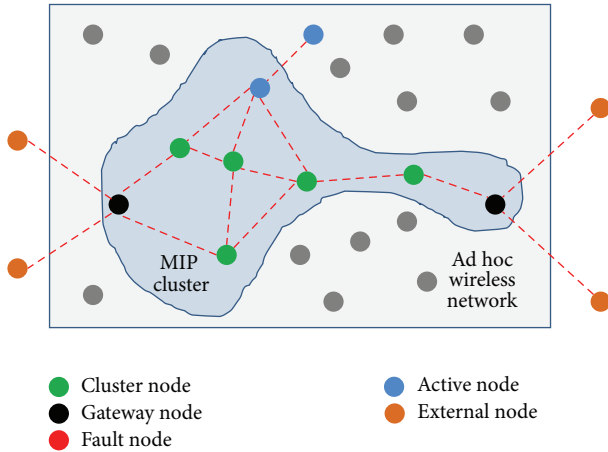


FIGURE 1: IP cluster built into the wireless ad hoc network.

wireless ad hoc network where at least one cluster has already been created. This wireless ad hoc cluster provides us with the transport infrastructure to deliver multimedia data from a source node to a destination node. They can be both internal and external to the ad hoc wireless network. Figure 1 describes the components of this scenario. Gateway nodes (GN) are defined in the MWAHCA architecture to be used as a bridge between two different networks or with external nodes. A GN is connected with the wireless ad hoc network by a wireless interface card, but they are also connected to another external network with a different interface card. This external network can be wired or wireless, but in this research we only use the wireless connections option.

When a cluster node has to transmit multimedia packets to another cluster node, it looks for the best route according to the routing algorithm provided by the MWAHCA architecture. This algorithm examines the whole cluster topology, which is defined by the assigned MIP, and it weights up the resources and characteristics of the nodes into the path in order to select the optimal multimedia route. When the best path is calculated, then, the required resources for the multimedia streaming are reserved by exchanging messages between the source and destination nodes, before starting sending multimedia packets. All nodes in the path are checked to guarantee that they have enough capacity and no more multimedia connections are allowed in a node when bandwidth resources run out.

If the routing algorithm offers different paths between source and destination nodes, only the best route will be selected according to the topology characteristics and the multimedia metrics. When a node on the selected route cannot make an appropriate resource reservation to accomplish the request from the source node, it sends a notification message back to the source node; thus it could eliminate this saturated node from the topology used by the algorithm to avoid bottlenecks and traffic congestion. Then, a new path is calculated and the resource allocation is tested again. Because routing algorithm works independently for each flow, this mechanism provides load balancing between two nodes into the same cluster. Sending different multimedia flows between

two nodes by different routes keeps latency, jitter, and lost packets under the thresholds criteria established by the MIP configuration.

When the resource allocation is successfully completed and multimedia streams are sent through the same path and the same nodes, thus jitter and latency will be the lowest because of the routing algorithm calculations. But the problem comes when a node in the path is disconnected, failed down, or when it just disappears from the cluster. Neighbor nodes will detect the fault node when it stops sending messages. Then, they update their state tables and send this change of topology to other nodes in the same cluster. When the source node finds out that there is a multimedia path that is not valid anymore because of a change in the topology, it stops sending multimedia packets and runs the routing algorithm again. If there is an alternative path, and the nodes meet the requirements, then a new route is built between the source and destination nodes. Consequently, the MWAHCA architecture can be considered inherently fault tolerant because of its capacity to adapt when changes happen. However, the architecture does not provide any specific fault tolerant mechanism; thus the convergence time needed to recover the multimedia flow can be high enough to break the multimedia communication. Even when the transmission is kept, the QoS parameters and the quality perceived by the end user will be really harmed in this time interval.

The convergence time depends on many factors: topology size, amount of nodes into the cluster, average adjacencies by a node, average distance between nodes, and node settings such as timers. From previous studies performed by us through simulations and real scenarios [10, 13], the convergence time can be estimated and it takes values in a wide range of values, on general between one and sixty seconds. For some applications, such as Internet Protocol Television (IPTV) or Video on Demand (VoD), convergence time values around a few seconds can be admissible because the multimedia transmission will be lost for a while but it will be restored quickly. A fault node will cause some sort of degradation in the quality of service perceived by the final user, but the transmission will not be interrupted. Nevertheless, if there is a real time bidirectional communication, like a VoIP call or a video conference between two or more people, and the destination node stops receiving the audio or video signal only a few seconds, the communication breaks down. Moreover, the user may end the transmission because he/she cannot hear or see anything.

Another different problem that we can face with this fault tolerance behavior is the suboptimal routing. If a new route is calculated and the multimedia transmission is restored, when the failed node restarts and it comes back to the cluster, the best path will not be calculated again and the available resources can be underused.

The related problems with the original behavior bring us to design a new complementary algorithm valid for any wireless ad hoc network in general and for MIP clusters based on the MWAHCA model in particular. In the next subsection this proposal is detailed.

3.2. Fault Tolerant Mechanism Based on a Fast Switching Path.

Our objective is to design a fault tolerant mechanism that should be able to restore a multimedia transmission path in less than a second. Moreover, this mechanism will be applied to individual unidirectional flows; thus if two or more flows are affected by a node failure, each one of them will be recovered separately. A bidirectional multimedia communication, VoIP call or video conference, is seen by this algorithm as two different unidirectional and independent flows. To achieve this objective, the algorithm establishes a temporary route called *fast switching path* (FSP). A specific FSP will be assigned to each affected flow to quickly switch from the broken path to a new optimal route when it is calculated. The purpose of this FSP is to reduce the convergence time, but at the same time to keep QoS parameters inside an admissible range of values. Multimedia transmissions usually use UDP protocol at the transport layer; thus when a node fails, it is not possible to completely avoid the packet loss because the acknowledge packets are not sent to confirm the correct packet reception. Hence, even with convergence time values lower than a second, some quality degradation will be always expected.

When a node is transmitting multimedia packets and the next hop in the path stops sending messages back to it, the node becomes an *active node* (AN). On one hand, the role of an AN seems similar to a source node, because it has to run the routing algorithm to calculate the best path from this point to the destination. But, on the other hand, an AN does not make a reservation because there is already a multimedia transmission in progress and it needs to restart sending packets as soon as possible. When the algorithm determines the new next hop to the destination node, then the active node sends a *Warning Message* addressed to the destination through this hop. Each hop between the active and the destination node that reads the *warning message* updates its information table and forwards the message to the new hop. The *warning message* is only used to advise the nodes about the new path to retransmit the packets as long as the resource starvation does not happen. Consequently, in the first stage of the fault tolerant mechanism, there will be two concatenated paths for the same multimedia flow in the same cluster. First path is part of the original broken route and it goes from the source to the active node; this is a guaranteed path because all nodes keep the resource allocation initially performed. The second path goes from the active to the destination node and it is the fast switching path. FSP does not have any guarantee of service and packet loss may happen, but destination will keep receiving packets just some milliseconds after the node failure has occurred.

In order to better understand how the proposed mechanism works, we are going to study an example topology. Figure 2 shows two multimedia transmissions along the same MIP cluster. In this example topology there are twelve nodes, two of them are GNs and they allow the ad hoc wireless network to connect with external nodes. The system is designed to provide multimedia transmissions between internal sensor nodes, between internal and external nodes or vice versa, or between external nodes. In this example we will describe the worst case, between external nodes. So,

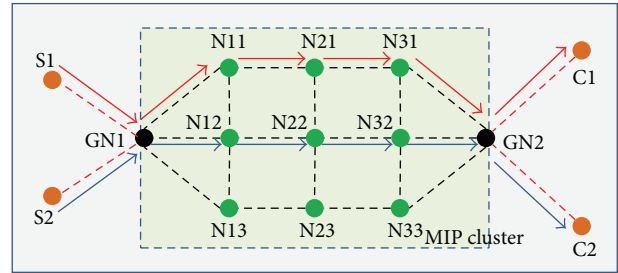


FIGURE 2: Two multimedia flows crossing along an ad hoc network using the optimal path.

Figure 2 shows how multimedia transmissions go from the left to the right external nodes. The wireless ad hoc network is only used as a transport network between external nodes. Thus, all the multimedia traffic that will be considered in this example crosses the ad hoc network from the gateway node 1 (GN1) to the gateway node 2 (GN2) because only a unidirectional flow has been taken into account. Some multimedia applications require bidirectional flows; however, as the fault tolerant mechanism is applied independently to unidirectional flows, this scenario holds all the necessary components to understand how it works. Moreover, it could represent a real case study where the wireless ad hoc network is used to deploy a multimedia service, extending their scope and coverage. Figure 2 also shows the best path calculated by the routing protocol for both multimedia flows when all nodes into the cluster are working properly, the adjacency processes have finished, and the network has converged. All nodes in the ad hoc network have the same characteristics, the same available resources, and they have been previously configured with the same MIP settings. Therefore, the main criterion used by the routing algorithm is to minimize the number of hops between GN1 and GN2.

Figure 3 shows the FSP calculated for the red flow when N31 node fails down. In the described scenario a node failure is simulated by shutting down the N31 node (the red node in the diagram). In a certain moment, packets of the red flow transmitted by the N21 node are lost. When N21 node breaks the adjacency with N31 node, it sends a *Cluster State Update (CSU) message* to the remaining neighbor nodes. Next, with the purpose of restoring the multimedia transmission quickly, the active node starts the fault tolerant mechanism running the routing algorithm to find out the FSP. Looking at the diagram, we can see that the best option to get to the destination is using N22 as next hop. Then, N21 sends to this node a *warning message* and the N22 node will retransmit the *warning message* to the N32 node, and this one to the GN2 node, who is the final destination inside the MIP cluster. Immediately after the active node has sent the *warning message*, multimedia packets transmission for the red flow is started again, but now it is done through the N22 node. The established fast switching path allows the system to

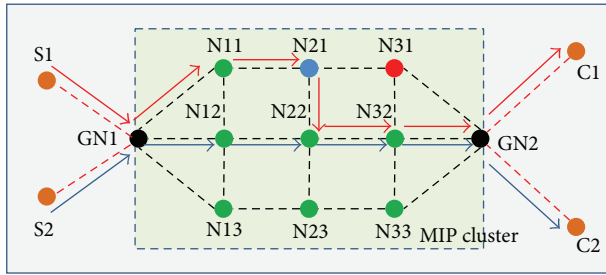


FIGURE 3: Fast switching path (FSP) recovery after a node failure.

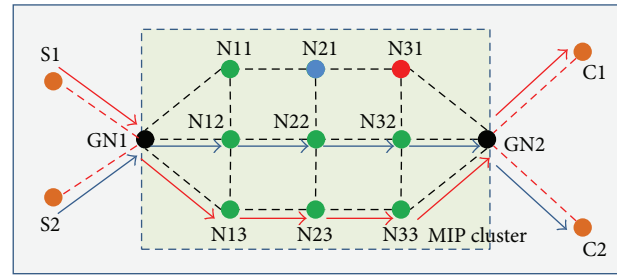


FIGURE 4: Restored path after the node failure.

keep delivering packets to the destination while minimizing the number of lost packets and the amount of time that the service was interrupted because of the node failure. Fast switching moves the system to a transitional state, where the main objective is to provide a continuous delivery of packets although the quality of service of these packets cannot be completely guaranteed. In this transitional path, as observed in Figure 3, the path between source and destination gateway nodes is not the optimal route because it has been modified by the active node to avoid the failed node. Furthermore, another inconvenient for this transitional route is that the restored multimedia flow is crossing several cluster nodes that are already managing some other multimedia flows, like the blue flow. Thus, if the fast switching path is kept during a long time, it can contribute to make a bottleneck and the network could become congested. In the topology shown at Figure 3, red and blue flows are being retransmitted through N22, N32, and GN2 nodes. This is the optimal route only for the blue flow which also has a valid resource reservation in these nodes. But red flow does not have any reservation between the N21 and the GN2 node; thus when one of these nodes runs out their bandwidth resources, the red multimedia packets will be dropped. Blue multimedia packets will not be harmed because the resource reservation for this multimedia flow is kept at all times.

Given the aforementioned reasons the transitional path should be changed as soon as possible to a new stable path. In order to know the optimal route, the path must be calculated at the source node of this flow (or the first hop in the MIP cluster if the source node is an external node), and not in the active node. It can be done, because the active node sends a *fault message* back to the source gateway node asking for a new route calculation, but in this process multimedia packets keep being sent through the transitional route provided by the FSP. Moreover, when the source gateway node receives the *fault message* it keeps sending multimedia packets through the current route until a new route is calculated by the routing algorithm and the resource reservation of this final route is performed. Figure 4 shows the new optimal route, called *restored path*, for the red multimedia flow, where the number of hops has been minimized (underused nodes were preferred). For

the nodes in this new route, a normal resource allocation has been performed following the MWAHCA architecture specifications and the original developed protocol [11, 13]. Finally, when the resource reservation for the restored path is completely confirmed, the source gateway node stops sending red multimedia packets over the FSP and starts sending them along the restored path. Additionally, a message is sent to the active node to release the assigned resources in the nodes in this path.

**3.3. Protocol Extensions.** With the proposed fault tolerance mechanism, MWAHCA architecture has not been modified and the processes and messages defined at the network protocol remain identical. This was one of the initial objectives of this proposal. The other one was to reduce the recovering time when a node fails. It will be demonstrated later on a theoretical study first and in a performance study in a laboratory test bench.

The fault tolerant mechanism can be incorporated to the existing system by an extension of the protocol. To achieve it, we need to define several new types of messages and some messages exchange processes. Because of the space limitation, only the new type of messages to achieve fault tolerance are introduced and exposed in this paper. The remaining types of messages and processes are used as defined in [10]. Table 1 shows the new types of messages definition for the fault tolerant mechanism.

MEDIA\_INFO structure holds information about the source and destination multimedia flow, the required resources, and the original path. Figure 5 shows the messages exchange has taken place between the affected nodes when a specific node fails in the cluster. When the routing algorithm selects the neighbor node, only the *warning message* is sent before the multimedia streaming packets are sent again from this new path. If the neighbor node has a valid route to the final destination, and this route does not include the active node, then it sends an *ACK warning message* to the active node to confirm the new route. When this happens, the active node will keep sending multimedia packets. The active node does not wait for the *ACK warning message* reception; thus the latency time for the buffered packets in

TABLE 1: Fault tolerant messages.

MESSAGE	TYPE	FAULT TOLERANCE MECHANISM		
		LENGTH	VALUE	DESCRIPTION
WARNING	0x41	13–41 bytes	NODE_ID MEDIA_INFO	Message to inform the next nodes that a multimedia flow without guaranteed services is going to be sent
ACK WARNING	0x42	13–41 bytes	NODE_ID MEDIA_INFO	Confirms the warning message when there is a valid route from this point to the destination
REJECT WARNING	0x43	13–41 bytes	NODE_ID MEDIA_INFO	Reject the warning message when there is not any valid route from this point to the destination
END_FAULT_TRAN	0x44	13–41 bytes	NODE_ID MEDIA_INFO	Releases resource reservation and notifies the previous hop in the path when a node fails
NODE FAULT	0x45	13–41 bytes	NODE_ID MEDIA_INFO	Notification of a node failure from the active node to the source node
ACK NODE FAULT	0x46	13–41 bytes	NODE_ID MEDIA_INFO	Confirm the node failure message sent to the active node

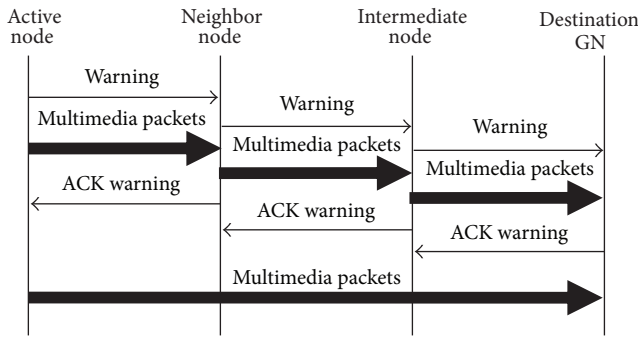


FIGURE 5: Fast switching path stage messages exchange.

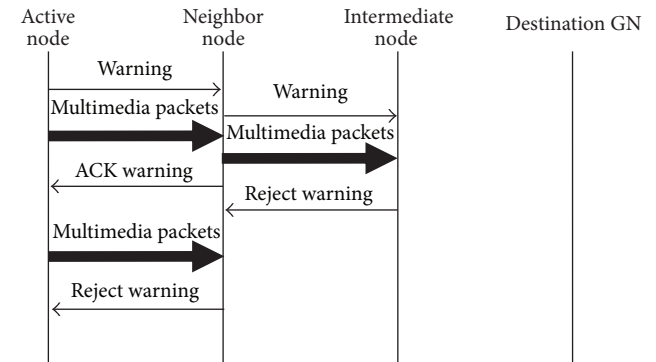


FIGURE 6: Fast switching path (FSP) fails because of resource starvation.

active node is significantly reduced. The same process is repeated in every hop until the destination GW is reached. The neighbor, intermediate, and destination gateway nodes keep the flow information held into the MEDIA INFO structure in a separated table of flows with a valid resource reservation; therefore, when the node becomes congested, it starts dropping packets from no guaranteed flows. When the gateway destination node starts receiving multimedia packets again, it stops the oldest path because it is not valid anymore. The *end fault transmission message* is used instead of the original *end transmission message* in order to notify the nodes in the path that the transmission is ending because of a node failure.

Figure 6 shows the messages exchange when a node in the FSP does not have enough available resources because it runs out its bandwidth. This mechanism is implemented to avoid the bottleneck formation. The intermediate node, which has its resources exhausted, stops forwarding the *warning message* to the next hops. Instead of that, it answers with a *reject warning message*. Then, the previous node stops forwarding multimedia packets and forwards the *reject warning message* back to the path. When the *reject message*

arrives to the active node, the multimedia packet stops until a new alternative FSP is calculated by the routing algorithm. If there is no valid alternative FSP from the active node to the destination node, the active node definitely ends the multimedia transmission by sending an *end fault transmission message* to the source node by using the original optimal path. In this case, the source node starts the *forward process* to find a new route to the destination.

As stated above, the new FSP is created only to keep the quality of service parameters as good as possible, but the active node has to inform the source gateway node that the current path has failed and a new optimal route has to be recalculated. A *node fault message* is sent from the active node to the source node with the MEDIA.INFO structure of the multimedia flow that must be restored. Figure 7 describes this scenario where the source GN finds a new optimal alternative route to the destination GN and immediately starts transmitting multimedia packets. This new path will be a guaranteed route because all intermediate nodes will have confirmed the resource reservation with a *reserve resources*

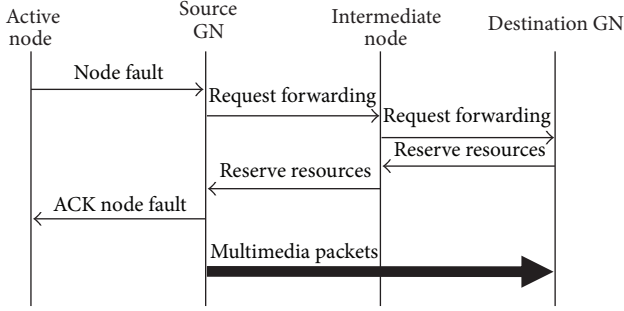


FIGURE 7: Recovery route procedure.

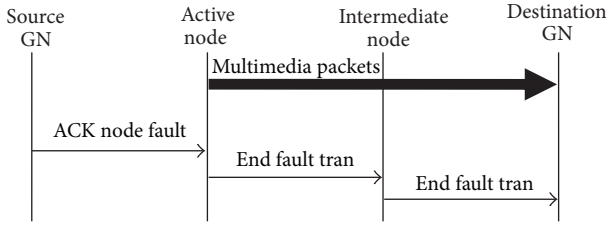


FIGURE 8: Procedure of resource release on a failed path.

message. The source GN node will send an *ACK node fault* message to the active node in order to notify it that the temporary fast switching route should be disassembled.

When the active node receives the *ACK node fault* message, it stops forwarding multimedia packets and it sends an *end fault tran* to the destination node in order to release the flow information from their tables. Figure 8 shows how the fast switching path is unmounted.

#### 4. Analytical Model of Convergence Time and QoS Parameters

In this section, the convergence time for the fault tolerant mechanism described above is described analytically. Then, the quality of service parameters is analyzed to know the impact of a failed node in the quality of multimedia transmission perceived by the end user.

We define the convergence time ( $T$ ) as the time interval between a node into the cluster which fails and a new route to the destination which is established. We assume  $n_a$  hops between the source node into the MIP cluster and the active node, the node before the failed node. The latency time to the active node ( $L_a$ ) is defined as the time needed by a message sent by the source node to reach the active node. The latency between the source node and the destination node into the cluster (with a distance of  $n_d$  hops) is defined as  $L_d$ .  $L_w$  is the time needed to send a *warning message* from the active node. Then, we can estimate the convergence time from two conditions: when the fault tolerant mechanism algorithm is not used,  $T_1$ , and when it is used,  $T_2$ . Then

$$\begin{aligned} T_1 &\geq L_a + 2L_d, \\ T_2 &\geq L_w. \end{aligned} \quad (1)$$

If we assume that the processing time is the same in both the source and active node and we make the approximation of taking the latency as a linear function of the number of hops, we can express  $L_a = n_a \cdot L$ ,  $L_d = n_d \cdot L$ , and  $L_w = L$ . Following these approximations, we can calculate the difference between the convergence time in the two considered conditions. It is estimated from (1) that

$$\Delta T = T_1 - T_2 \approx L_a + 2L_d - L_w = (n_a + 2n_d - 1)L. \quad (2)$$

If we call  $R$  the average number of packets sent by a specific multimedia flow per second, then we can calculate the average packet loss ( $P$ ) in the two aforementioned conditions:

$$\begin{aligned} P_1 &\geq (L_a + 2L_d)R, \\ P_2 &\geq RL_w. \end{aligned} \quad (3)$$

And following the same approximations than in (2), the difference between lost packets when we use the fault tolerant mechanism is given by

$$\Delta P = P_1 - P_2 \approx (L_a + 2L_d)R - RL_w = (n_a + 2n_d - 1)RL. \quad (4)$$

In order to have a picture of the behavior of the proposed model, we will show how the system performs in this model for the cases of  $n_a = \{2, 3, 4\}$ . We will take  $n_d = 4$  and we assume  $R = 400$ . The convergence time and number of lost packets can be represented as a function of  $L$ . Figures 9(a) and 9(b) show the linear function of  $\Delta T$  and  $\Delta P$  when  $L$  rises. The improvement achieved on reducing the convergence time and the number of packets will be bigger when more delay introduces a single node. Moreover, higher  $n_a$  provides higher  $\Delta T$  and  $\Delta P$  values. Thus, results can be especially interesting on WSN where nodes have really limited resources.

#### 5. Performance Study

In this section, the measurements obtained from the test bench are presented. The fault tolerant mechanism described in this paper has been incorporated to the protocol explained in [10]. The protocol extension has been programmed in JAVA language. Figure 10 shows the topology used in these tests. Each dashed line matches the adjacency between two different nodes. The yellow node, N22, represents a congested node where the available resources have been reduced by other multimedia flows. This node has a large number of adjacencies. The paths between the gateway nodes are longer when they cross the N22 node. Black nodes are gateway nodes; GN1 is connected to the external node S1, which is running the VLC software as a video streaming server, and GN2 is connected to the external node C1, which is running the same VLC software but as a client of video streaming receiving the S1 signal. Green nodes are cluster nodes with identical characteristics; all of them have the same amount of available resources. The only criterion used by the routing algorithm is the number of hops because of the laboratory settings and the node configurations. Only a multimedia unidirectional flow is being transmitted from S1 node to C1



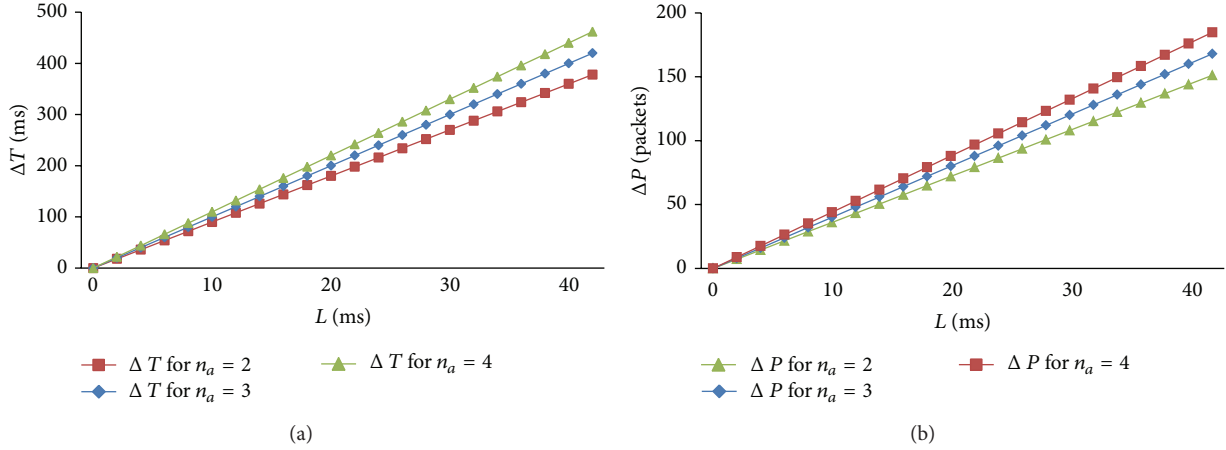


FIGURE 9: (a)  $\Delta T$  as function of  $L$ , (b)  $\Delta P$  as function of  $L$ .

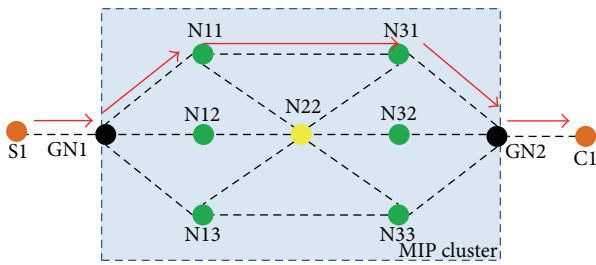


FIGURE 10: Experimental topology.

node. When transmission starts in S1, multimedia packets are flooded to GN1 where the developed protocol is used. GN1 extracts the multimedia information into the IP, UDP, and RTP headers to build the MEDIA\_INFO structure. Then, the routing algorithm calculates the optimal route to reach the GN2 node, which is the destination node into the ad hoc cluster. The N22 node is discharged because the minimum amount of hops is four while distance along the upper path, GN1-N11-N31-GN2, and the lower path, GN1-N13-N33-GN2, is only three hops. There is no load balance implemented; therefore the routing algorithm selects the route whose next hops have a lower NODE\_ID. In Figure 10, it is N11 node. In summary, if the video streaming starts after the MIP cluster has converged, the route selected will be GN1-N11-N31-GN2, with three hops. We will call this initial route as *baseline path*.

Once, all nodes in the MIP cluster are working properly, we gathered measurements of latency, jitter, and packets loss at C1. They represent the optimal values for this specific network arrangement because the optimal path has been established when all nodes are working and no other multimedia flow or IP traffic is on the ad hoc network. Measured values for QoS parameters depend on the bandwidth consumption spent by the selected video codec. Latency, jitter, and lost packets values of a video streaming with a higher bandwidth consumption codec will also be higher. In order to validate the results, independently of the codec features, three different codecs have been selected in this study: 600 Kbps, 1.500 Kbps, and 3.000 Kbps.

While C1 is receiving video streaming, a node failure is introduced in the network by forcing to shut down N31 node. Following the fault tolerant mechanism introduced in this paper, the N11 node becomes active node when the N31 failure is detected. Thus, it looks for an alternative path through the other neighbor node, N22. We called this temporary route as *fast switching path*. It consists of the optimal path from GN1 to N11 concatenated to the new temporary route calculated by the routing algorithm between N11 and GN2. The whole *fast switching path* is GN1-N11-N22-N32-GN2. The expected results are that QoS values in this stage must be worse than *baseline path* values because this is not the optimal route and no resources reservation has been performed on N22, N32, and GN2 nodes. However, if fault tolerant algorithm is working properly, the video streaming will not be interrupted at any time.

At the final phase, the recovery mechanism tries to find a new optimal path between GN1 and GN2. When the active node, N11, sends the *fault node message* to GN1, a new forwarding process starts in the source gateway node. Looking at the diagram of the topology we can see that the new optimal path will be composed by the following nodes: GN1-N13-N33-GN2. This third and definitive path is called *recovered path*. Then, measurements of latency, jitter, and lost packets are performed using a video streaming of 600 Kbps, 1.500 Kbps, and 3.000 Kbps in three different conditions.

- (i) *Baseline path*: average values are calculated from the last 100 multimedia packets before the N31 node fails.
- (ii) *Fast switching path*: average values are calculated from all packets received by C1 in this stage. While this is a temporary path, only a limited number of packets (usually less than one hundred) will be sent along this route.
- (iii) *Recovered path*: average values from the first 100 multimedia packets received by the C1 node have taken 10 seconds after the N31 went down. A 5-second waiting time is established to ensure that the MIP cluster topology has converged and the fault tolerant mechanism has already established the recovered

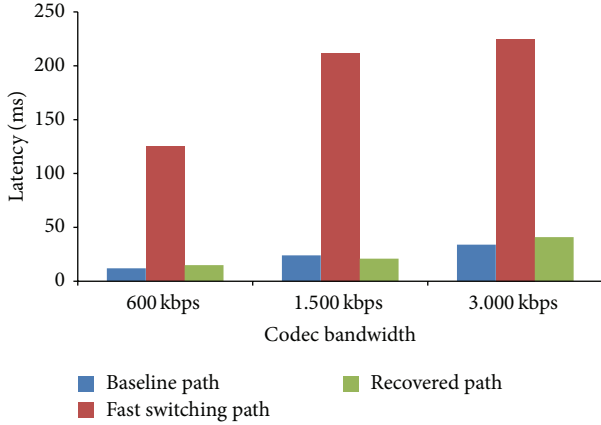


FIGURE 11: Latency in different stages of the fault tolerant process.

path properly. This value has been calculated from preliminary tests.

Figure 11 shows the average latency values in different stages of the fault tolerant process. We can see how the average latency is similar when we compare the baseline and 10 seconds after the node failure happens, in the recovered path stage. This result demonstrates that the multimedia transmission has been recovered and the latency values in the new optimal route are similar to the values achieved before the failure. On the other hand, the average latency in the fast switching path is significantly higher than in the other two paths. This result has a logic explanation because the temporary path built in the fast switching state is longer (4 hops in the ad hoc network) than the initial and final path (3 hops). We must also consider that some packets were buffered at the active node while the routing protocol was running. Latency for individual packets in this phase ranges between 100 and 300 ms, depending on the used codec. From these results we can conclude that latency values are valid for multimedia transmission, either unidirectional or bidirectional flows, in the three experimental conditions because of the fault tolerant mechanism.

Figure 12 shows the average jitter measures. As it happened with the latency, jitter in the baseline and recovered path are similar and, at the same time, they are considerably lower than in fast switching path. Nevertheless, in this case, jitter values in the temporary path have values that could be considered too much high for a real time bidirectional multimedia communication. This negative effect can be successfully eliminated in a unidirectional transmission, like a video streaming, by setting up a jitter buffer in the node at the end of the communication, for example, in the gateway destination node, just before leaving the MIP cluster or, even, at the end user device, CI. But, in a bidirectional real time transmission, like Voice over IP (VoIP) or video conference, the jitter buffer cannot take high values without harming the latency values.

Figure 13 shows the average lost packets percentage measured in each phase with identical criteria than in the previous cases. One more time obtained results confirm the

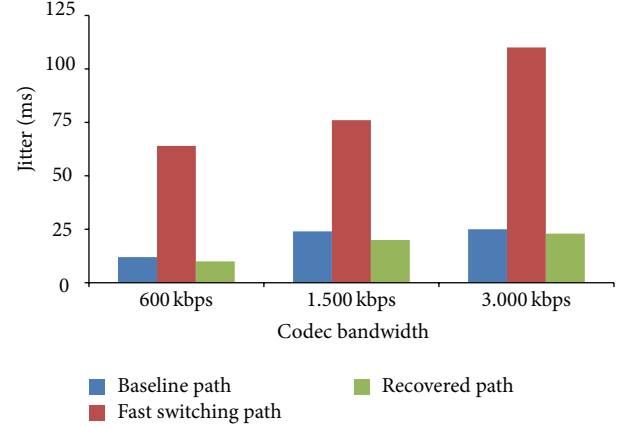


FIGURE 12: Jitter in different stages of the fault tolerant process.

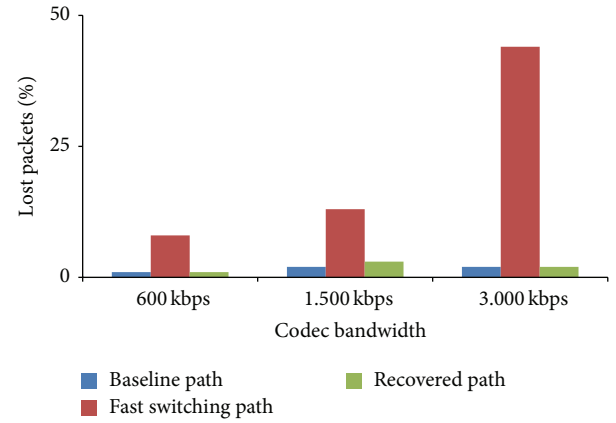


FIGURE 13: QoS parameters in a fault tolerant process with a 3.000 Kbps codec.

multimedia flow has been completely restored when the fault tolerant mechanism is implemented. In this particular case, lost packets in the codec with higher bandwidth requirements, 3.000 Kbps codec, show an excessively large number of lost packets (nearly 50 percent in the fast switching path).

Finally, with the same test bench and the same settings, we studied the average convergence time in four different experimental study cases.

- (1) The proposed fault tolerant mechanism is not running and, moreover, N31 node is shut down. This means that the original protocol is used and N31 node sends a message to N11 node before it fails down. Thus, N11 node knows the topology change before the *baseline path* is interrupted.
- (2) The fault tolerant mechanism is not running but, now, N31 node is suddenly shut down. Then, the original protocol is used and N31 node does not send any message before it fails down. The *baseline path* is interrupted before N11 node knows the adjacency with N31 is not valid anymore.
- (3) The developed fault tolerant mechanism is running and N31 node is shut down. Then, the fault tolerant

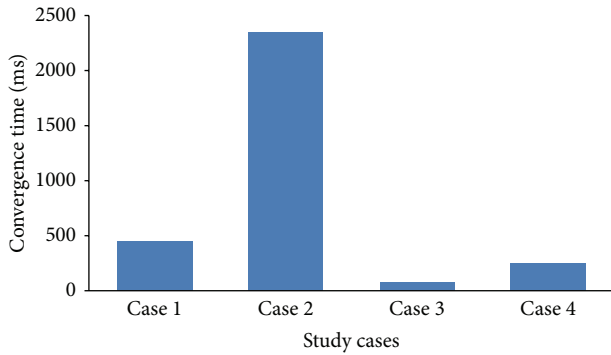


FIGURE 14: Convergence time.

features proposed in this paper are incorporated to the protocol and N31 node sends a message to N11 node to break the adjacency; thus N11 becomes an active node before the *baseline path* is lost.

- (4) The developed fault tolerant mechanism is running and N31 node is suddenly shut down. Then, the proposed fault tolerant features have been incorporated to the protocol and N31 node does not send any message before failing down.

Figure 14 shows the convergence time for each one of the above study cases. Convergence time has been calculated as the difference between the time of arrival of the last packet before lost packet happens and the first packet after the multimedia transmission is restored over an optimal and guaranteed path. Results show an evident difference between these cases. The fault tolerance is used in cases 3 and 4 and cases 1 and 2 do not use the proposed algorithm. The fault tolerant mechanism achieves a convergence time lower than half a second when the proposed algorithm is used.

## 6. Conclusion

In this paper, we have studied the fault tolerance characteristics of the MWAHCA architecture and protocol. Several shortcomings have been detected; thus a new fault tolerant algorithm has been proposed to be applied over ad hoc wireless networks. This mechanism uses a temporary *fast switching path* to improve the QoS and QoE parameter in the recovery transition. In this process multimedia packets are forwarded through a new route, which may not be the optimal route and some nodes on the path have not properly performed a resource reservation. With the new fault tolerant mechanism, convergence time is significantly reduced and the amount of lost packets is minimized. While multimedia packets keep being forwarded through the fast switching path, the fault tolerant mechanism builds a third and optimal path, the *recovered path*, in order to improve other QoS parameters such as latency or jitter. The last path is built following the original forwarding process provided by the MWAHCA architecture; therefore resource reservation is made in all nodes on the path. Results in the laboratory show that the fault tolerant algorithm works as expected. It

allows a multimedia flow to keep crossing the ad hoc network with a minimum convergence time after a node fails. The measured QoS parameters indicate that both unidirectional and bidirectional multimedia services will not be interrupted at any time. Nevertheless, two relevant restrictions must be considered, mainly in the FSP.

- (i) Jitter should be reduced by configuring a 20–60 ms buffer at the end user side.
- (ii) Lost packets values for high bandwidth consumption codec can reach an excessive value.

On the other hand, the low time when the FSP is used jointly with the achieved low convergence time makes this fault tolerant mechanism improve the multimedia communication.

In the future we want to improve the jitter and the lost packet results in multimedia flows by adding high bandwidth requirements. Another research line is to implement load balancing in ad hoc networks with high density of devices. Moreover, our purpose in the near future is to add security to our system through the use of network topology discovery algorithm [24].

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work has been partially supported by Instituto de Telecomunicações, Next Generation Networks and Applications Group (NetGNA), Covilhã Delegation, and by National Funding from the FCT-Fundação para a Ciência e a Tecnologia through the Pest-OE/EEI/LA0008/2013 Project.

## References

- [1] M. Tarique, A. Hossain, R. Islam, and C. A. Hossain, "Issues of long-hop and short-hop routing in mobile ad hoc networks: a comprehensive study," *Network Protocols and Algorithms*, vol. 2, no. 2, pp. 107–131, 2010.
- [2] S. Sendra, J. Lloret, M. García, and J. F. Toledo, "Power saving and energy optimization techniques for wireless sensor networks," *Journal of Communications*, vol. 6, no. 6, pp. 439–459, 2011.
- [3] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection and verification," *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.
- [4] J. Lloret, I. Bosch, S. Sendra, and A. Serrano, "A wireless sensor network for vineyard monitoring that uses image processing," *Sensors*, vol. 11, no. 6, pp. 6165–6196, 2011.
- [5] M. Garcia, D. Bri, S. Sendra, and J. Lloret, "Practical deployments of wireless sensor networks: a survey," *Journal on Advances in Networks and Services*, vol. 3, no. 1-2, pp. 170–185, 2010.
- [6] S. A. E.-F. Ali and S. K. Refaay, "Chain based fault tolerant routing protocols," *Network Protocols and Algorithms*, vol. 4, no. 3, pp. 79–103, 2012.

- [7] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.
- [8] D. Rosário, R. Costa, H. Paraense et al., "A hierarchical multi-hop multimedia routing protocol for wireless multimedia sensor networks," *Network Protocols and Algorithms*, vol. 4, no. 4, pp. 44–64, 2012.
- [9] J. Lloret, M. Garcia, D. Bri, and J. R. Diaz, "A cluster-based architecture to structure the topology of parallel wireless sensor networks," *Sensors*, vol. 9, no. 12, pp. 10513–10544, 2009.
- [10] J. R. Diaz, J. Lloret, J. M. Jimenez, and J. J. P. C. Rodrigues, "A QoS-based wireless multimedia sensor cluster protocol," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 480372, 17 pages, 2014.
- [11] J. R. Diaz, J. Lloret, J. M. Jimenez, and S. Sendra, "MWAHCA: a multimedia wireless Ad Hoc cluster architecture," *The Scientific World Journal*, vol. 2014, Article ID 913046, 14 pages, 2014.
- [12] A. S. Sadiq, K. A. Bakar, K. Z. Ghafoor, and J. Lloret, "An intelligent vertical handover scheme for audio and video streaming in heterogeneous vehicular networks," *Mobile Networks and Applications*, vol. 18, no. 6, pp. 879–895, 2013.
- [13] J. R. Diaz, J. Lloret, J. M. Jiménez, and M. Hammoui, "A new multimedia-oriented architecture and protocol for wireless Ad Hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 16, no. 1, pp. 14–25, 2014.
- [14] E. Pagani and G. P. Rossi, "Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks," *Mobile Networks and Applications*, vol. 4, no. 3, pp. 175–192, 1999.
- [15] G. Gupta and M. Younis, "Fault-tolerant clustering of wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking (WCNC '03)*, pp. 1579–1584, New Orleans, La, USA, March 2003.
- [16] A. T. Tai, K. S. Tso, and W. H. Sanders, "Cluster-based failure detection service for large-scale ad hoc wireless network applications," in *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 805–814, Florence, Italy, July 2004.
- [17] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Fault-tolerant clustering in ad hoc and sensor networks," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS '06)*, p. 68, Lisboa, Portugal, July 2006.
- [18] Y. Xue and K. Nahrstedt, "Providing fault-tolerant Ad hoc routing service in adversarial environments," *Wireless Personal Communications*, vol. 29, no. 3-4, pp. 367–388, 2004.
- [19] A. Boukerche, R. Werner Nelem Pazzi, and R. Borges Araujo, "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments," *Journal of Parallel and Distributed Computing*, vol. 66, no. 4, pp. 586–599, 2006.
- [20] T. Bheemarjuna Reddy, S. Sriram, B. S. Manoj, and C. Siva Ram Murthy, "MuSeQoS: multi-path failure-tolerant security-aware QoS routing in Ad hoc wireless networks," *Computer Networks*, vol. 50, no. 9, pp. 1349–1383, 2006.
- [21] H. L. Chao and C. L. Chang, "A fault-tolerant routing protocol in wireless sensor networks," *International Journal of Sensor Networks*, vol. 3, no. 1, pp. 66–73, 2008.
- [22] R. Melamed, I. Keidar, and Y. Barel, "Octopus: a fault-tolerant and efficient ad-hoc routing protocol," *Wireless Networks*, vol. 14, no. 6, pp. 777–793, 2008.
- [23] T. Antoniou, I. Chatzigiannakis, G. Mylonas, S. Nikolettseas, and A. Boukerche, "A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range," in *Proceedings of the 37th Annual Simulation Symposium*, pp. 43–52, Washington, DC, USA, April 2004.
- [24] P. Lopes, P. Salvador, and A. Nogueira, "Methodologies for network topology discovery and detection of MAC and IP spoofing attacks," *Network Protocols and Algorithms*, vol. 5, no. 3, pp. 153–197, 2013.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

