An Efficient Fault Tolerant Scheme for Mobility Management in Wireless Networks

Abel DIATTA¹, Ibrahima NIAANG¹, and Mandicou BA²
¹Département de Mathématiques et Informatique, Laboratoire d’Informatique de Dakar (LID)
²Département Génie Informatique, Ecole Supérieure Polytechnique (ESP)
Université Cheikh Anta Diop, Dakar, Sénégal

Abstract—Mobile communications are nowadays highly developed thanks to the multiplicity of mobile devices. Some communications pass directly between mobile nodes (because the latters have direct connections between them), while for others, the mobile node must pass through a point of attachment (PoA). In the latter case, when the PoA falls down, all mobile nodes that are attached to him losing communication with their counterparts.

In this paper, we develop a way to avoid this loss of mobile nodes communications even when their PoA falls down. This, thanks to an algorithm that we propose to strengthen the capacity of the Media Independent Handover Function (MIHF) in terms of managing the handover, especially during a failure of a PoA. In other words, our algorithm combines management of the continuity of communication during handover and managing the fault tolerance of the PoA.

Keywords: Fault Tolerance, Mobility management, Wireless Networks, MIHF

1. Introduction

With the proliferation of mobile devices (smartphones, tablets, laptops, ...), wireless networks have become essential nowadays. Indeed, the ability of users to communicate, send files, ... while moving, do that users are genuinely interested over these networks. In most cases, the communications are via applications using P2P networks (Skype, viber, WhatsApp ...) [1]. These two opportunities (possibility of movement during communication, use of P2P technology) offered to the user, are problematic. Indeed, in P2P technologies you need a good strategy for (i) fault tolerance because the nodes arrive and depart at any time. Meanwhile, in an environment characterized by high mobility of nodes, it is essential to take into account the (ii) frequent disappearances of links, but especially (iii) communications interruptions and loss of packets due to change coverage areas (handover) or disruption of an Attachment Point (PoA).

In the existing literature, work that included fault tolerance [2], [3], [4], [5] do not integrate mobility communication into their work. In other words, in these works, two users who are communicating are forced to stay on one place until the end of their communication. This is a heavy constraint. In the same vein, research works that tried to manage mobility nodes [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], did not take into account the fault tolerance.

Yu Liu et al. in [13] used a P2P technology to manage mobility. However, their study focused on how to avoid interference between the nodes. Abhishek Dhiman et al. in [14] treated the vertical and horizontal handover but they were interested in the throughput and delay (when a node moves at a given speed from one Access Point (AP) to another or from one Base Station (BS) to another) in Wi-Fi and WiMAX. They not only did not address the continuity of communication between two mobile nodes, but they have especially not integrated the possibility that a BS or an AP goes down while mobile nodes attached to it are in communication (ie fault tolerance). It is the same for papers [16], [17], [18], [19] in which the authors conduct studies on the performance of applications such as FTP, Video conference ... in WLANs, WiMAX, UMTS in terms of delay for the handover but not also, in terms of traffic sent and received. In [20], autors ensure the continuity of service during handoff but not take into account the fault tolerance.

In this paper, we propose a solution to ensure the continuity of communication between two mobile nodes even when their PoA falls down i.e we integrate both mobility management and fault tolerance.

The rest of the paper is organized as follows. In section II, we give the related works on Mobility and fault tolerance in Wireless mobile Networks. In section III, we give our contribution. Section IV gives details on our solution for Mobility and Fault Tolerant Management in Wireless mobile Networks. A performance analysis of our solution is done in this section. Section V is devoted to the conclusion and our future works.

2. Related Work

2.1 Survey on Mobility in Wireless Networks

In [8], authors, about searching for files have set up a cluster system based on mobile Ad hoc networks approach. Their solution is mainly based on the creation of clusters head (CH) and secondary CH. It presents notorious limits on the managing of CH failures. Indeed in their approach, the secondary CH must wait a time $t$ seconds, if it does not
receive messages from the CH, it considers it down and takes over. Thus, in their approach, a member of a cluster will wait $2 \times t$ seconds ($t$ for CH and $t$ for secondary CH) before conﬁrating that its CH is down and that it will seek another CH. In a ﬁle-sharing system, this is tolerable. By cons in the case of real-time communications that is unacceptable. By cons in the case of real-time communications that is unacceptable.

In [9], the solution established by Jabbar et al. allow mobile devices to connect themselves without a point of attachment (Wi-Fi Direct). Therefore, in their solution, mobile nodes are to be conﬁned within a small geographical area. What is not suitable for internet or social networks.

In [10] Kim et al. have set up a system to ensure the handover but only in the context of data loss. Their solution avoids data loss by selecting a peer agent to store the data of the mobile node (MN) which is moving to another area. Once it will be connected to another point of attachment (PoA), peer agent transmits it the data it had guarded and the transmission continues. In their approach, when the MN enters into handover, it does not receive data until it establishes a connection with another PoA. In other words, during this time the communication is interrupted. In the case of video transmission, as in their case the solution is relevant. However, in the case of communication via Skype calls, viber ... for example, this is unacceptable.

Kuo et al. in [11] and Angoma et al. in [20] have provided solutions to ensure service continuity during the handover. However in [11] they took into account only the horizontal handover. While in [20], their deployment in a real environment has not taken into account the possibility that a PoA fails during communication or that the access point is moved in the case of Wi-Fi. It is very possible especially now with the existence of wireless routers.

In [12], authors considered the horizontal and vertical handover (HHO and VHO). However, their study was limited mainly to show the impact of the movement speed of the mobile not only in terms of packet loss but also the terms of time required for handover. They have not implemented a strategy to ensure the continuity of communication during handover.

In [21], authors use the packet retransmission system for managing fault tolerance. Indeed if after some time an ACK is not received, they retransmit the packet. However, it should be noted that their solution does not solve the problem when the cluster is down. Because we can retransmit the packet as many times as we want, it will always be the same scenario.

In [22], Zayara et al. based their study on the comparison in terms of signal strength and handover (HO) delay. They applied the comparison of the two types of network integration namely the loose coupling and tight coupling but also on WiFi and WiMAX networks. To manage the HO, authors set up a system which, when the link between MN and PoA is lost, the Media Independent Handover user (MIH user) initiates the discovery of a candidate network. The MN checks the RSS (Received Signal Strength) and the bandwidth of the WiMAX network. If the bandwidth is greater than a threshold deﬁned in the MN, then MN starts the execution of the HO. The problem with their solution is that if there are several WiMAX networks that cover the area and whose bandwidth is greater than the threshold set in the MN, there will create a conﬂict because the MN will attempt to connect to all these networks at the same time.

2.2 Related Work on Fault Tolerance Mechanisms

In [2], [3], messages ping / pong are used to verify the breakdown of nodes. In fact, if after some time, a node does not respond with a message pong, it is declared down. We know that the response time is strongly dependent on the quality of the network. Otherwise, in their solution, a node can be declared out when it is not (simply because the response has been slow to happen due to poor network quality or a temporary disconnection of the link).

According to [4], a system in which the degree of distribution (degree of connectivity) is high, is more vulnerable to attack. But conversely, it provides a much more efﬁcient communication and better fault tolerance. By cons, a system where the degree distribution is not strong is more resistant to attack, but less effective in terms of communication and fault tolerance. So be in the middle (ie a high degree of constant distribution). This is what Suto et al. in [4] wanted to manage by implementing the hub nodes with a high degree and non-hub nodes with a low degree. However, their solution does not solve the problem. Indeed their degrees depend on the total number of nodes in the network. However, in networks with the size of the internet, characterized by high Chuns, the number of nodes in the network continuously changes. By applying their solution, the system will be unstable. Sometimes it has better fault tolerance and therefore more vulnerable to attack because the degree became high, sometimes communication is lacking because the degree is again low (because of several departures of nodes).

In [6], Lun et al. have developed a method for detecting failures of nodes. To do this, they put up a message storage tree (ms-tree). Each node sends a message to others. Each node receiving the message saves the message source in the ms-tree; in turn sends the message to others, and then adds the source node in a NFLP list (Non-Faulty-Like Peer). They realize the transmission message during three steps and after that, each node counts the number of times each node appears in the list NFLP. If a node appears a number of times less than $n - \left\lfloor \frac{n - 1}{3} \right\rfloor$, then this one is considered down ($n$ is the number of peers in the network). A large incoherent in this system is that in highly dynamic systems (arrival and departure at any time), a node can arrive in the system at
the third step of sending the message. It is clear that in this case, it will appear in the NFLP a number of times less than \( n - \left\lfloor \frac{n-1}{3} \right\rfloor \), and therefore will be declared down while it is not.

3. Contribution

We propose a model combining both the management of mobility and fault tolerance of nodes (MNs and PoAs) based on a hierarchical wireless network. Unlike flooding message used, our routing system avoids overloading the network because each PoA, by sending a message to its neighbors, precise to them in a list, all other neighbors to which it sent the same message. Therefore, the later PoA will not send the same message at the same latters even if they are its neighbors.

In addition, our solution helps strengthen the MIHF (Media Independent Handover Function) protocol by adding new features including those to continue communication even when the PoA falls down. This, due to the use of priority and backup addresses.

4. An efficient Fault Tolerant Solution for Mobile Wireless Networks

4.1 Basic Idea

The idea of our solution is very simple. We started with the following conclusion:

1) All the papers that have dealt fault tolerance [2], [3], [4], [5], [23]
   • consider that a node is down, if it is physically defective or does not respond to a ping message that was sent to it,

2) All the papers that have worked on the mobility of nodes in wireless networks [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] did not take into account the possibility that a PoA falls down during one of its son communicates

Considering all this, our solution sets a time \( t \) after which if a node does not respond, the source node asks two of its neighbors if they can contact the destination. If their answer is "NO", then the later is considered broken. By cons, if at least one of the two answers with "YES" this means that it is the connection between the source and the destination which is a problem.

This verification is especially important that declare a node as failed while it is not, creating unnecessary additional operations. In fact, if a node is declared down, all the nodes under its responsibility (especially when it is a backbone) will undertake updates to their routing table when it was not necessary since the node always work.

In our solution a node can be considered as a PoA, if at least \( k \) nodes can connect to it but also if it has a fixed IP address. PoAs are linked together randomly. The mobile nodes are connected to different PoAs via Wi-Fi network, WiMAX, UMTS, etc. Whenever a MN connects, its distance from the PoA is relieved. Unlike a lot of work, when this distance changes (ie when the MN moves), the HO is triggered, in our case, when the absolute value of this distance increases, the PoA by flooding sends a message to its neighbors to tell them that it has a son which wish to connect to them. This message contains the MN information.

The MN connects to the first PoA that responds and then informs its former PoA that it is connected to such PoA. Each PoA contains a list of MNS that are attached to it. In addition, each PoA contains information about its neighbors (distance, bandwidth, ...).

Drawing inspiration from [14], a MN has multiple IP addresses according to networks to which it belongs (Wi-Fi, WiMAX, UMTS,...). However, there is an address that is marked as a priority. This is the address obtained in the network to which the MN is attached. To find which network the MN is attached, we compare the absolute values of the distances between the MN and the PoA. MN will be attached to the PoA with which the absolute value is the smallest. It is this address that will be considered as a priority until the MN moves to another network. Figure 1 shows our hierarchical wireless architecture.

![Fig. 1: Hierarchical Wireless Networks](image-url)

4.2 Architecture operating

4.2.1 Routing

In our architecture, a MN wishing to contact another, sends a call message (audio, video ...) to its PoA. The message contains the information of the source and the destination. The PoA checks whether the destination is not attached to it. If attached to it, the message is sent directly to the MN, otherwise the PoA sends, by flood, the message to all its neighbors. In this message, the PoA precises the list of all neighbors to whom it sent the message. This will prevent other neighbors to send the same message to those which...
have already received (to avoid overloading the network). This list will be re-initialized at each PoA. The PoA checks whether the destination is not attached to it. If attached to it, the message is sent directly to the MN, otherwise the PoA sends, by flood, the message to all its neighbors except those which have already received, so on (see Algorithm 1).

For sending the response, we use the same scenario. The destination (ie the source of the response) sends the response to the PoA. This later verifies if the MN destination of the response (ie the source of the original message) is not attached to it (because after sending the message, it can move to another PoA). If attached to it, it sends it the response, otherwise the PoA sends the response to all its neighbors, stating in the message the list of other neighbors to which it also sent the response. Each of the neighboring check in turn if the MN destination of the response is attached to it. Otherwise, it sends the response to all its neighbors except those that have already been traversed by the response. The same process continues until the MN destination.

To send the response, the same algorithm is used. We simply reverse the roles of $MN_{src}$ and $MN_{Dest}$.

The worst case complexity of this algorithm is obtained on lines 12, 13 and 14. Let $n$ be the number of PoA.

For each PoA, we must go through the list of its neighbors. Therefore the complexity is a function of $n \times$ (size of each PoA neighbors list). These instructions will be in the worst case until all PoA receive the message. In other words, the complexity is $n \times n \times$ (size of each PoA neighbors list). However, the size of the PoA neighbors list is a most equal to connectivity degree of that PoA.

Let $k$ be the maximum connectivity degrees of PoA. So complexity is $O(k \times n^2)$. We recall that $n$ is the number of PoA and not the total number of nodes.

### 4.2.2 Node departure

For node departures, if it is a MN that is leaving the network, it informs its PoA and then leaves. The PoA removes it from the list of MN that are attached to it. However, in the case of it is a PoA that leaves the network (e.g. an AP), it sends information of its neighbors (relative distances, bandwidth, ...) to every MN which are attached to it, informs its neighbors and then leaves. Each MN attempts to connect to the PoA closest to him.

The complexity in the worst case is obtained when it is a PoA leaving the system. This should send a message to all its sons. So the complexity depend on the (number of MNs attached to PoA). In addition, the PoA leaving the system inform all its neighbors. Therefore the complexity is also dependent on the (number PoA neighbors). Then, the complexity depends on the (number of MNs attached to PoA) + (number of PoA neighbors). This is at most equal to the degree of connectivity of the PoA.

Let $k$ be the connectivity degree of PoA. Therefore the complexity is $O(k)$.

### 4.2.3 Node joining

When a MN arrives, it sends a broadcast message. In the message, it specifies that it wants to connect to a PoA. If there is a PoA which responds, it tries to connect to it. The response contains the PoA information thereof (position, bandwidth, number of remaining connection, ...). If multiple PoA respond, the MN chooses to connect to the PoA which has the largest bandwidth. If they have the same bandwidth, the MN then connects to the nearest PoA.

If it is a PoA which arrives, it sends a broadcast message stating its position, bandwidth. Each PoA receiving the message is considered as its neighbor. The latter responds to arriving PoA stating in turn its position, bandwidth, etc. It saves it in its neighbor list. The arriving PoA also recorded in its neighbor list, all the PoA that responded.

---

**Algorithm 1: Sending msg from $MN_{src}$ to $MN_{Dest}$**

1: send($MN_{src}$, msg, idPoA) \% MN sends msg to its PoA
2: if ($MN_{Dest}$ ∈ List$_{PoA}$(MN)) then
3: send (idPoA, msg, idMN$_{Dest}$)
4: else
5: foreach PoA neighbor
6: send (idPoA, msg, List$_{PoA}$(idNeighbor))
7: end for
8: repeat
9: if ($MN_{Dest}$ ∈ List$_{PoA}$(MN)) then
10: send (idPoA, msg, idMN$_{Dest}$)
11: else
12: foreach PoA neighbor
13: if ((idPoA $\in$ List$_{PoA}$(idNeighbor)) and
14: idPoA $\notin$ List$_{PoA}$(Neighbor)))
15: send (idPoA, msg, idPoA)
16: end if
17: end for
18: until (($MN_{Dest}$ ∈ List$_{PoA}$(MN)) ∨ (all PoA receive the msg))
19: end if

---

**List: 
- $MN_{src}$: MN source
- $MN_{Dest}$: MN destination
- List$_{PoA}$(MN): List of MNs attached to PoA
- idMN: MN Identifier
- idPoA: Identifier of PoA source
- idPoA$_{i}$: Identifier of PoA $i$
- num.neigh: Number of neighbors
- List$_{PoA}$(Neighbor): List of PoA neighbors
- List$_{msg}$_{PoA}$(Neighbor): List of neighbors receiving the msg**

**Function:** send(source, msg, destination)
Complexity is based at most on the number of degree of connectivity (ie the number of neighbors of incoming PoA and the number of MNs attached to it). Therefore the complexity is \(O(k)\).

4.2.4 Fault-tolerance

As we said above, the mobility management works consider that a node is down when it does not respond to a "ping" message during a time \(t\). But the arrival of a message (request or response) depends on other factors such as the quality of bandwidth, link status, etc. That is why in our case, all nodes send to their neighbors (MN or PoA) ping messages at regular time interval. When a node does not respond to a "ping" message for a time \(t\), the source asked two of its neighbors if they can contact the destination. If their answer is "NO", then the later is considered broken. All its neighbors suppress it in their neighbor list. By cons, if at least one of the two answers is "YES" this means that it is the connection between the source and the destination which is a problem.

When a node fails, the following steps are performed:

1) If it is a MN that is down:
   - its PoA removes it from its MNs list
   - its neighbors suppress it in their neighbors list

2) If it is a PoA that is down:
   - each MN connects to the PoA of its backup address
   - each PoA which was connected to the down one removes it from its neighbor list and attempts to connect to another PoA

The complexity in the worst case is obtained when it is a PoA that fails. Each node that was connected to the latter changes its neighbor table by removing the failed PoA in the neighbor list. Let \(k\) be the connectivity degree of PoA. There are at most \(k\) nodes that were connected to the failed PoA. Each of these \(k\) nodes must go through the list of its neighbors \(k \times \) (size of list)). The size of this list is at most equal to \(k\). Therefore the complexity is \(O(k^2)\)

4.2.5 Managing Continuity of communication during handover

Handover management includes three stages [22]:
- Initiation of handover: When the MN moves
- Available networks Discovery
- Execution of handover: Connect to an available network

In papers which worked on the management of handover, when the MN moves, the handover is triggered (ie the implementation of the Media Independent Handover Function (MIHF)). This sometimes creates unnecessary operations (thus overloading the network). Because the MN can move by approaching more its PoA, and thus its signal becomes better. It is unnecessary in this case to trigger a handover.

In our case, we trigger the handover if and only if the absolute value of the distance of the MN with its PoA increases (as this shows that the MN moves away from its PoA).

To implement our strategy for ensuring the continuity of communication, we are inspired by [14] where authors use Master IP.

When a node integrates the network, the address that was provided to it in its PoA network, is considered as a priority. If the MN straddles several other networks, it will have other addresses it got from other PoA. Among these addresses, the one it obtained from the PoA which has the largest bandwidth will be marked as a backup address. What will serve this backup address?

Response: When the PoA of the MN (ie the one of its priority address) is faulty, backup address is used directly to continue the communication. Hence, we avoid the interruption of communication.

In summary, in our solution, we trigger the handover in two cases:
- If the distance between the MN and its PoA grows
- When the PoA of at least one of the communicating MN fails (even if the MN does not move).

This is summarized in the following algorithm:

**Algorithm 2: Continuity of Communication Management**

1: If distance (MN, PoA) grows
2: trigger MIHF
3: Else if PoA falls down \(\%\) MN does not move \%
4: MN connects, meanwhile, to 2\(^{nd}\) priority network
5: MN checks the best network in terms of bandwidth
6: If best network <> from 2\(^{nd}\) priority network
7: MN connects to the best network
8: End if
9: End if

We have added a new feature in the MIHF (Media Independant Handover Function), these are lines 3 to 9 of algorithm 2. Indeed, some fault tolerance of a PoA was previously not included in the MIHF. In other words, before our algorithm, a failure of a PoA causes the breakdown of communication.

Similarly, the complexity depends on the number of PoA neighbors to which was attached the communicating MN. Let \(k\) be the connectivity degree of PoA. The MN in communication is connected to the PoA. Therefore the number of other PoAs which are neighbors of the latter PoA is at most equal to \(k - 1\). Since this is one of those \((k - 1)\) PoA neighbors that the communicating MN seeks an alternate; the complexity is therefore \(O(k - 1)\).

4.3 Performances Analysis

In this section, we first give a summary table of the costs of the different algorithms we have implemented, then we make a criticism of our solution.
• **FT**: Fault Tolerance
• **MCC\textsuperscript{CHO}**: Managing Continuity of Communication during Handover
• Let \( K \) be the set of degrees of connectivity of all PoAs
• \( k = \text{Max}(i), \ i \in K\)
• \( n \): number of PoA

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Routing</th>
<th>Arrival nodes</th>
<th>Nodes departure</th>
<th>FT</th>
<th>MCC\textsuperscript{CHO}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>( O(k \times n^2) )</td>
<td>( O(k) )</td>
<td>( O(k) )</td>
<td>( O(k^2) )</td>
<td>( O(k - 1) )</td>
</tr>
<tr>
<td>Memory Occupation</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>low</td>
</tr>
</tbody>
</table>

Table 1: Complexity of our algorithms

The core strength of our solution is that it is very suitable for scaling. The more the number of nodes are, the higher the degree of connectivity of the system is. Thus, for operations such as routing, for example, there are more routes to which the messages are sent. Thus, it becomes faster to find a destination.

However, the main disadvantage of our solution is that it takes up too much memory space especially in the routing. Because each PoA must consult two lists: its neighbors list and the list of these neighbors which have already received the message.

5. Conclusion and future works

In this paper, we have set up a management architecture of both mobility management, fault-tolerance and ensuring communication continuity in wireless mobiles networks. Most solutions that use the flooding routing are major polluters of bandwidth. By cons, in our cas we have given a routing algorithm (algorithm 1) which, although it uses flooding messages, avoids overloading the network. This is due to the sending by each PoA to its neighbors, the list of other PoAs to which it sent the message. In addition, we have given another one (algorithm 2) which, if embedded on a wireless network adapter of a mobile node, will allow it to continue communication even in case of failure of the PoA.

In the case of our very close perspective, we will focus on an experimental simulation analysis first and then by deployment.

References


[20] B. Angoma, M. Erradi, Y. Benkouaz, A. Berqia, and M. C. Akalay,

