Distrib Parallel Databases (2016) 34:101–117
DOI 10.1007/s10619-014-7159-5

Merging sub-networks in self-managed vehicular ad-hoc networks

Cándido Caballero-Gil · Pino Caballero-Gil · Jezabel Molina-Gil

Published online: 16 October 2014

Abstract The most widespread wireless technology for mobile ad-hoc networks nowadays is Wi-Fi based on the IEEE 802.11 standard. The working procedure of this protocol produces various problems when multiple sub-networks are merged. The main drawbacks that may cause inability to communicate are the IP duplication and the existence of sub-networks in different channels. This paper proposes a practical solution to these problems, which does not cause any network overload and does not affect the data management of connections. Furthermore, it has been fully implemented in a new tool developed to create a vehicular ad-hoc network using only smartphones. Both a straightforward deterministic algorithm and a more complex scheme taking into account the interferences between wireless channels from a fuzzy logic point of view are defined here. Promising results regarding performance and security have been obtained from the analysis of large-scale NS2 simulations based on data got from implementations with real devices.

Keywords Sub-network · VANET · IEEE 802.11 · Large-scale simulation

1 Introduction

A Vehicular Ad-hoc NETwork (VANET) may be seen as a form of Mobile Ad-hoc NETwork (MANET) that is used to provide communications between vehicles in

C. Caballero-Gil · P. Caballero-Gil (✉) · J. Molina-Gil
Department of Computer Engineering, University of La Laguna, 38271 Tenerife, Spain
e-mail: pcaballe@ull.es

C. Caballero-Gil
e-mail: ccabgil@ull.es

J. Molina-Gil
e-mail: jmmolina@ull.es

order to increase the comfort and safety of all road travel. The IEEE 802.11 standard is used in most existing MANETs to deploy Wi-Fi connection because it enables short-to-medium-range wireless communication capability (up to 300 m). The traditional definition of VANET suggests connections through the variant of such a standard known as IEEE 802.11p, which was specifically designed for vehicle communications. That proposal assumes the existence of on board units (OBUs), which are installed inside vehicles, and roadside units (RSUs), which are installed in the road. However, these devices are not yet available to the public. This research proposes the use of the IEEE 802.11b/g communication protocol instead of IEEE 802.11p because the first standard is already present in all current smartphones, but the second one is not yet available in almost any device. Thereby, this proposal will enable a fast deployment of a VANET working on the IEEE 802.11b/g protocol, because this is present in most current smartphones.

In IEEE 802.11 networks, each device first checks whether some wireless network exists in its neighbourhood or not. Then, if the network exists the device tries to connect to it, and otherwise, the device creates a new wireless network. The existence of multiple instances of a traditional wireless network in real environments can cause problems mainly due to the spontaneous formation of sub-networks.

The two most important issues in this case appear when two or more sub-networks cannot see each other because they are on different channels, and when two or more sub-networks use the same channel but there are devices with the same IP addresses. The first problem usually appears because the choice of the wireless channel is random. Moreover, the problem of duplication of IP has an easy solution, which is connected with the solution proposed here to the problem of merging sub-networks.

The most common types of broadcasting channels in the IEEE 802.11xx protocol are a, b, g and n. Its range is from channels 1 to 11 in America, 1 to 13 in Europe and 1 to 14 in Japan, and in all cases channels share part of their bandwidth. Figure 1 shows the channels that the devices can use in Europe to create a wireless network, remarking the maximum number of channels that can be used without any interference. In particular the maximum number of simultaneous wireless networks without interference is five.

The different versions of the IEEE 802.11xx protocol provide the advantage of being compatible with each other, so that the user does not need anything more than its integrated Wi-Fi adapter to connect to the network. However, too much saturation in a channel may cause packet collisions in the data transfer, which would correspond to a lower transfer rate and/or speed. The ideal situation is that existing sub-networks are on the same channel and without any IP address conflict in order to merge them with no problem. However, this case is unusual spontaneously.

This work proposes practical solutions to the two aforementioned problems, which have been implemented in a new tool developed to create a VANET using only mobile phones with Wi-Fi. Smartphones with the IEEE 802.11 protocol provide the advantage

Fig. 1 Channels

of being compatible with each other, so that the user does not require anything more than its integrated Wi-Fi adapter to connect to the network.

This paper is structured as follows. The following two sections briefly describe related work and our contribution. Sects. 4 and 5 introduce respectively a generic deterministic proposal and an approach based on fuzzy logic. Sects. 6 and 7 include the analysis of performance and security. Finally, sect. 8 provides some conclusions.

2 Related work

Many research projects and bibliographic references exist on organization of MANETs [4] and VANETs. On the one hand, the paper [18] deals with a problem similar to that addressed here, because it proposes solutions to the data management of concurrent execution of transactions, but in MANETs instead of VANETs. On the other hand, the work [11] uses large-scale self-organized ad-hoc networks where nodes are divided into sub-networks. Also, the authors of [2] present a self-organizing data life cycle management of MANETs, while the paper [9] addresses the issue of building real-time traffic information system in VANETs. Another work [6] proposes cooperative groups to avoid false traffic warning messages in VANETs. While most studies on this topic are mainly theoretical, some of them use simulations based on tools such as NS2 [2], or show implementations in real environments [5]. Two different methods are here proposed that combine large-scale software simulations with data obtained from implementations in real devices.

Several works face different problems of the IEEE 802.11 protocol. The authors of [8] include a performance analysis of the 802.11b protocol using analytical modelling. The work [10] studies the relationship between clock synchronization and energy-saving in IEEE 802.11 MANETs. The paper [12] discusses the accuracy of IEEE 802.11 signals in indoor location. The authors of [13] describe and simulate an algorithm to mitigate the Bluetooth interference with the channel estimation stage on the IEEE 802.11g. All previous proposals deal with problems of IEEE 802.11, but none of them follows the approach of this paper, where solutions to practical problems are proposed.

Other authors discuss the sub-network merging process when an IP address conflict problem exists, [7]. The paper [17] tries to solve several problems related to channels in those ad-hoc networks where multiple channels are used simultaneously. The work [1] shows one way to estimate and avoid the noise in the channel by using fuzzy logic, which is a brief approximation to the problem. The paper [19] proposes a framework for multi-radio multi-channel cognitive wireless networks to, which allows maximizing a network utility function based on data routing, resource allocation and scheduling. The work [15] studies the allocation of non-overlapping channels in wireless mesh networks, minimizing interference, enhancing the capacity of the network, and maintaining the connectivity of the network. The same problem in mesh networks is discussed in the paper [14], which proposes a dynamic centralized interference-based algorithm aimed to minimize interference. The work [16] studies multi-hop ad-hoc networks using conventional IEEE 802.11 where long transient resynchronization states usually occur The authors propose a modification of the resynchronization, which reduces times and energy consumption.

None of the above research papers deals with exactly the same problem presented in this work. Furthermore, they do not show the level of details of implementations with real devices, which is one of the highlights of this paper.

3 Our contribution

VAiPho (VANET in Phones) [3] is a new tool to create VANETs by using only smartphones inside vehicles playing the role of OBUs. It automatically detects in real time different types of events such as traffic jams and sends warnings about them to other smartphones via Wi-Fi in order to provide drivers with information about road traffic conditions. A fundamental requirement for this tool is that each mobile device is both client and server, and connects to the same wireless network. This network is especially suitable for urban environments where the vehicle speed is not so fast, but the scheme also works in vehicles on highways in the same direction and at junctions. However, for vehicles traveling in opposite directions at high speed, this scheme does not work.

In VANETs implemented through VAiPho, when two instances of the same wireless network exist on different channels, the so-called sub-network merging problem arises. The ideal solution would be the biggest sub-network absorbing the smallest one, but nodes do not know the size of the other sub-networks. Once both sub-networks are connected, communications on the selected channel grows. Then, there are proposals to reduce communications overhead that could be applied after the merging procedure. This problem is not taken into account because it is not the topic of this work. Data management of communications does not represent a problem in our proposal because the number of generated packets is very low compared to the amount of data that is typically broadcast over a wireless network in general.

A different problem is the data management of connections between nodes that are to be merged with other sub-networks. Since these nodes have to change their IP, they have to restart their connections. However, those nodes that had been previously authenticated can resume their connection quickly so that the only effect of the merging process will be a short delay. This work describes first a generic algorithm to compute the minimum time that any device in a sub-network needs to reset and connect with other sub-network, depending on the size of its first sub-network. Fuzzy logic rules are used to interpret interferences between channels in terms of sizes of neighbouring sub-networks in order to allow larger sub-networks to absorb the smaller ones. One of the key points of this paper is the performance analysis of large-scale NS2 simulations based on data obtained from implementations with real devices.

4 Deterministic sub-network merging

In order to solve the sub-network merging problem, this paper addresses the data management problem of replicated sub-networks by proposing the reset of the wireless interface of those devices belonging to all but one wireless sub-networks.

Algorithm 1 Choosing the sub-network with less interference

```
...
//several vaipho networks exist
int[] channelDifference = new int[nNetworks];
//channelDifference stores the difference between channels.
for (int i = 0; i < vaiphoInstances; i++)
  channelDifference[i] = 50;
for (int i = 0; i < nNetworks; i++)
  for (int j = 0; j < nNetworks; j++)
    if (network[i].Equals(wifi)) //compare sub-network with others
      if ((i != j) &&
      (Math.Abs(channel[i] - channel[j])) < channelDifference[i])
        channelDifference[i] = Math.Abs(channel[i] - channel[j]);
  //the channel with biggest difference between channels is not reset
  for (int i = 0; i < nNetworks; i++)
    if ((biggestDifference < channelDifference[i])
    &&(channelDifference[i]!=50)) //store the biggest difference
      biggestDifference = channelDifference[i];
      nodeLocationBiggestDifference = i;
    if (nodeLocationBiggestDifference != numberOwnNetwork)
    //if my network has not the biggest interference
      networkDetachProtocol(wifi);
      return true;
...
```

Fig. 2 Basic scheme

The best solution would involve resetting those sub-networks with fewer devices, but this information is not transparent to the devices of each sub-network, which can know only the number of devices in their sub-network. If a node detects an IP address conflict, it just has to reset its network interface, as this solves the problem. However, the problem is more difficult when sub-networks exist on different channels.

4.1 Basic scheme

A first approximation is based on the choice of the sub-networks that are in the most appropriate channel to minimize the interference created by the networks in the adjacent channels (see Fig. 2). The algorithms are written in C# because the implementation is for Windows Mobile. Each device running VAiPho regularly checks for *vaipho* sub-networks.

If there are two instances of the network called *vaipho*, all devices belonging to one of the sub-networks have to reset their wireless interfaces. The first node that detects the situation sends a warning message to all other nodes, so that they also restart their wireless interfaces. Thus, after receiving the message, the nodes broadcast it, turn off their network interface for 1 s, and then reactivate it. Then, since a *vaipho* network already exists when the nodes are restarted, they connect to this network and the problem is solved. Nodes remain authenticated by the nodes of the previous sub-network because authentication data do not vary with the change of sub-network, which keeps the same name, *vaipho*.

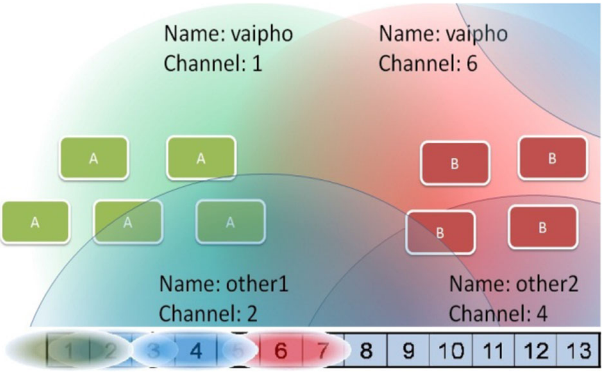

Fig. 3 Example

This method has the potential problem of mutual reconnection. If two devices that are on different channels simultaneously detect the existence of another sub-network, they could restart both sub-networks. However, in this case, the creation of two new sub-networks in the reconnection of devices is unlikely.

4.2 Optimized choice

When several devices belonging to different sub-networks are neighbours, each device can determine if there are other sub-networks. To allow the deployment of VANETs by VAiPho, each device has to check whether other *vaipho* sub-networks exist in its transmission range.

If there are different sub-networks with the same name because they were created independently in different channels, these sub-networks must merge. Each member of the sub-networks that discovers the problem must choose a new channel. of particular, the devices of the sub-network with the largest channel number restart their network interfaces so that the new channel of the combined sub-networks corresponds to the smallest number of all previous channels. However, this way to choose the sub-network that survives is not the best one because it does not depend on the number of nodes in each sub-network, or on the interferences between channels used by other wireless networks. Each device can know only a few parameters such as the number of nodes that are in its sub-network, the channel where it is, in which channels there are other instances of the *vaipho* network, and the channels that are being used by other networks that may interfere with its *vaipho* sub-network. The interference in channels can be used to determine the sub-network that will be restored.

Figure 3 shows a typical example of a VANET in an urban setting where two sub-networks of vehicles, denoted A and B, enter the same transmission range and find out two *vaipho* instances formed on different channels, and also interferences with other wireless networks. Each node of a sub-network that can see nodes of other sub-networks checks the channel where its sub-network is and the channels of the remaining sub-networks, and analyses possible interferences with other wireless

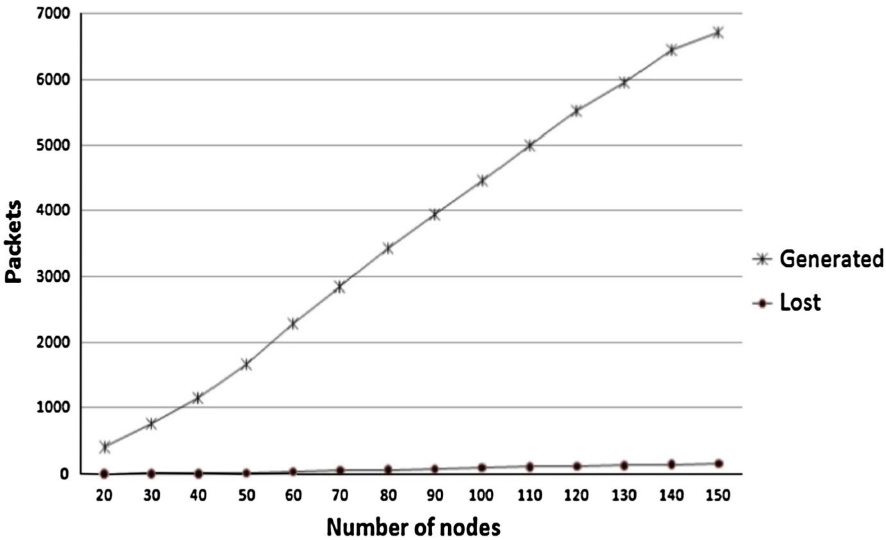

Fig. 4 Number of generated and lost packets

networks. In the figure, the sub-network A is on channel 1, B is on channel 6, and two other wireless networks are on channels 2 and 4. Consequently, A's channel, which is 1, is at a distance 1 from the nearest network, which is channel 2; while the channel of B, which is 6, is at a distance of 2 from the nearest network channel, which is 4. Thus, it is concluded that the channel of the sub-network A has more chance of interference, and consequently those devices must restart to merge with sub-network B. However, the best solution should be to restore the sub-networks with fewer devices so in the next section a new fuzzy logic approach is proposed to achieve it.

The second issue is the IP duplication. This problem is easily solved if the first node that detects the IP duplication resets automatically its network device.

5 Fuzzy logic based proposal

The number of generated and lost packets along a merging process is probably lower if the sub-network whose devices restart their network interfaces is the one with fewer nodes, as show Fig. 4. This figure shows data resulted of large scale simulation executed with data obtained from real experiments whose parameters are described in Sect. 6. Figure 4 also shows that the number of lost packets grows with the number of nodes, however, this number is inappreciable and can be ignored. This happens because vehicles in a traffic jam are evenly spaced, what makes that the signal is not saturated. From this starting point, we propose a heuristic procedure that, without knowing the number of nodes in other sub-networks, allows that the sub-network with fewer devices or more interference is the sub-network whose nodes restart their wireless interfaces. In addition, another important factor that is also taken into account is the interference with other existing sub-networks channels.

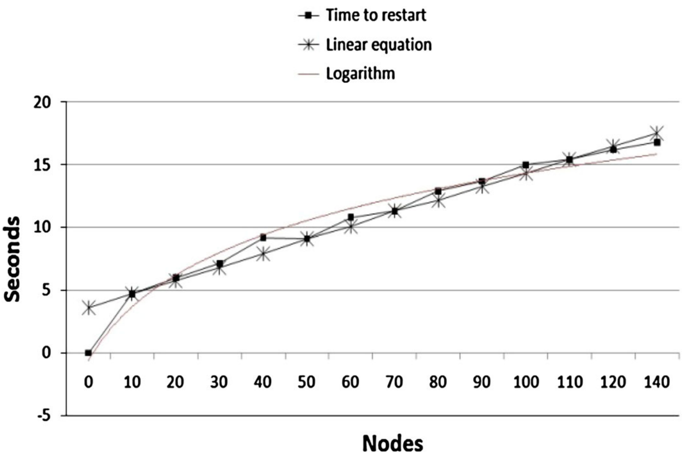

Fig. 5 Time relay for sub-network merging

In this paper, a fuzzy logic approach is used to estimate the time that each node waits before checking whether the problem remains unsolved, so, in that case, it begins the merging process. To estimate the time, not only the number of nodes in each sub-network is considered, but also a correction factor based on the interference between channels. Two targets of computing the waiting time are that the times should be minimized and that the nodes belonging to the chosen sub-network should not restart their interface because that would involve that all nodes would reset their interfaces.

$A(x)$ is the time that each node waits before checking whether it has to restart its network interface. It is expressed in terms of the number x that are the nodes belonging to its sub-network. For the fuzzy logic approach, times shown in Fig. 5 were computed with real devices. These times can be either linearly approximated by the Eq. 1, where Sr is the residual standard deviation that gives the average variability of the data about the regression line.

$$A(x) = 0.107x + 3.63 \pm Sr \quad if \quad x > 0 \tag{1}$$

Expression (1) has been used as starting points to estimate the average time before checking whether there are multiple *vaipho* instances. In particular, it is proposed the use of a linear expression by removing the initial constant time 3.63. Thereby, a node detecting multiple instances of the *vaipho* network waits for a while depending on the number of nodes in its sub-network $W(x) = 0.107x$. For example a node detecting several instances of the *vaipho* network such that its sub-network has 3 nodes, will wait for 0,321 s before rechecking if there are more than one *vaipho* instance, and there is another node who is detected in the *vaipho* network but belongs to a different *vaipho* sub-network of 20 nodes will wait for 2.14 s before rechecking. In this way, the smallest sub-networks would join the largest sub-networks.

There are two points to select the sub-network to restart. The first one is the number of nodes in each sub-network. The second one is the interference between channels that should be also considered so the sub-network with less interference prevails over others.

A sub-network operating on a channel is said not to have interference if no other network is on the same channel or in less than two channels away from it. The interference power of a channel c can be measured in *dBm*, which is a unit used to express the absolute power of a network signal with a power level P, through the logarithmic Eq. 2.

$$dBm = 10 \log \frac{P}{1mW} \tag{2}$$

This metric (Eq. 3) allows to estimate the interference $I(c)$ that a network using a channel c has. N denotes the number of sub-networks using channel c; M and Q denote respectively the numbers of sub-networks within 1 or 2 channels of distance from channel c, and a is got from the expression $\frac{2a}{dBm_{high}} = 1$ where dBm_{high} corresponds to the channel with highest interference.

$$I(c) = \sum_{i=1}^{N} \frac{-a}{dBm_i} + \sum_{j=1}^{M} \frac{2(-a)}{3dBm_j} + \sum_{k=1}^{Q} \frac{-a}{3dBm_k} \tag{3}$$

The above expression results from that if two sub-networks work on the same channel, the overlap is total. If they are in adjacent channels, the overlap is 2/3 of the frequency they use. If they are two channels away, the overlap is 1/3 of the frequency they use. The value of a could vary depending on the spot where vehicles are because if they are in a location with several wireless networks, the possibility of having more than two sub-networks using the same channel is higher.

At this point, the use of fuzzy logic to apply the interference between channels of sub-networks is proposed in order to estimate the waiting time for checking and resetting. Based on actual data from the experiments it was concluded that the expression that defines a degree of interference B for a *vaipho* instance dependent on the interference d(resulting from $I(c)$ in Eq. 3) of the own sub-network can be estimated according to the following equations, where Eq. 4 corresponds to the case with no interference with other wireless networks, and Eq. 5 corresponds to when there is interference with other wireless networks.

$$B(d) = \begin{cases} 0 & \textit{if} \quad d < -90 \\ 1 + 0.11d & \textit{if} \quad d \in (0, -90) \end{cases} \tag{4}$$

$$B(d) = \begin{cases} 1 & \textit{if} \quad d < -210 \\ -0.005(d+10) & \textit{if} \quad d \in (-210, -10) \\ 0 & \textit{if} \quad d \in (-10, 0) \end{cases} \tag{5}$$

The proposal to estimate the waiting time until the sub-network merging considers the degree of interference between two instances of *vaipho* network. Figure 6 shows an estimation of such a degree, which depends both on the power of the network and on the existence of interference with other networks.

Figure 6 illustrates the design of the fuzzy control rules, denoted by Rules 1 and 2, which depend on the interferences of the network of the node performing the test,

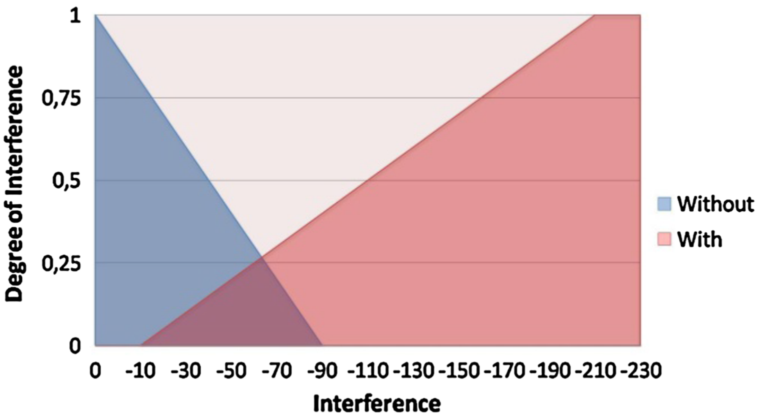

Fig. 6 Fuzzy logic approach of sub-network merging

Table 1 Rules

Rule 1 Own instance of the vaipho network
if ($I(vaipho(\alpha)) > -10$) then //With interference
$F(x) = W(x) + \frac{DI(vaipho(\alpha))}{2}$
else //Without interference
$F(x) = W(x) - \frac{DI(vaipho(\alpha))}{2}$
Rule 2 Other instances of the vaipho network
if ($I(vaipho(\beta(l))) < -90$) then //Without interference
$F(x) = F(x) - \frac{DI(vaipho(\beta(l)))}{2}$
else //With interference
$F(x) = F(x) + \frac{DI(vaipho(\beta(l)))}{2}$

denoted by $vaipho(\alpha)$; and of the other sub-networks, denoted by $vaipho(\beta(l))$ where $l \geq 1$.

$F(x)$ denotes the time that the x nodes of the sub-network $vaipho(\alpha)$ wait before re-checking whether other *vaipho* instances exist. According to the rules defined in Table 1, if a different *vaipho* instance exists after this waiting time, the detecting node begins the merging protocol. Otherwise, it does not restart its interface. Rule 2 is always checked after Rule 1 to determine the time that is added or taken away to the waiting time $F(x)$.

Every sub-network $vaipho(\alpha)$ that have x nodes estimates its own waiting time $F(x)$ by taking into consideration not only its own data but also another sub-network $vaipho(\beta(l))$ data. Figure 7 shows the waiting time a node awaits before rechecking *vaipho* interfaces, which depends on the amount of nodes of the corresponding sub-network and on the interferences in the channels of $vaipho(\alpha)$ and $vaipho(\beta(l))$.

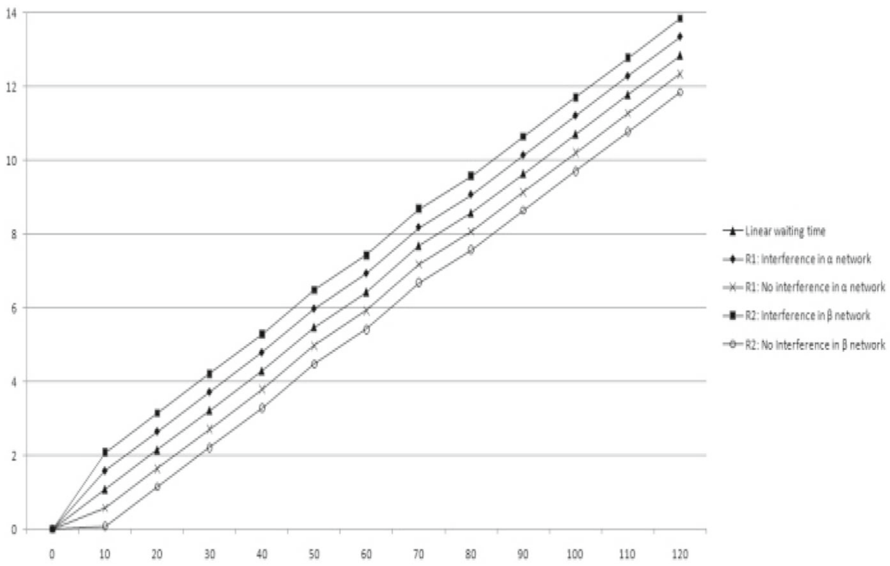

Fig. 7 Waiting time depending on the interferences

6 Performance evaluation

In order to check the performance of the proposal, the ideal option would be to check it with thousands of real devices, but this option is not feasible, so the alternative to simulate the scenery in a realistic way was to implement the scheme in a few real devices, getting actual data regarding reconnection and merging, and then using those data in a large-scale NS-2 simulation.

6.1 Real device implementation

The testing on real devices has been done on mobile phones with Windows Mobile 5 and 6 (see Table 2), and the development on Microsoft Visual Studio 2008. The reason to choose this type of devices instead of the latest devices with Android, iOS or Windows Phone operating systems is due to the fact that the companies behind these new devices decided to limit the access to the control of wireless networks interface, so nowadays, due to this current software limitation, Windows Mobile is the only platform where developers can completely control the wireless network interface. Anyway, it is planned that soon, a future version of Android will allow some of the key features of Windows Mobile that made possible the implementation presented here. If this happens, from then the implementation presented here will be also possible for Android devices.

Figure 8 shows a screenshot of two emulators using the same *vaipho* sub-network while another *vaipho* sub-network exists in another channel. The details of networks are shown on Fig. 9. In this example, the first device which detect the presence of more

Table 2 Mobile devices

Phone name	Platform	CPU speed	RAM	ROM	Battery capacity
HTC HD Mini	WM 6.5	600 MHz	384 MB	512 MB	1200 mAh
HTC P3300	WM 5.0	201 MHz	64 MB	128 MB	1250 mAh
HTC Touch2	WM 6.5	528 MHz	256 MB	512 MB	1100 mAh
hp IPAQ 614C	WM 6.0	520 MHz	128 MB	256 MB	1590 mAh

Fig. 8 Real device implementation

than one instance of the vaipho sub-network is the one on the right. When the node detects that there exist other *vaipho* instances in its range, starts the merging protocol, then, the node who detects the other sub-networks is responsible to broadcast a warning message to the rest of nodes which belong to the same sub-network. Later, these nodes will restart their network interfaces. Then every node turns off its network interface for 1 s, turns it on again, and automatically connects to the existing *vaipho* sub-network (Fig. 8).

In Fig. 9 a real state with different wireless networks in the same transmission range is shown. Apart from other network interfaces, there are two instances of the *vaipho* sub-network on channels 1 and 11. Therefore, it is necessary starting the protocol in order to merge both sub-networks.

SSID	Default Authentication	Default Encryption	RSSI (dBm)	Channel	Frequency (MHz)	Network Mode	Network Type
eduroam	WPA2/802.1x	AES-CCMP	-43	1	2412	802.11g	Access Point
vaipho	Open	None	-34	1	2412	802.11g	Independent
welcome@HTW	Open	None	-42	1	2412	802.11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-42	1	2412	802.11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-69	11	2462	802.11g	Access Point
eduroam	WPA2/802.1x	AES-CCMP	-69	11	2462	802.11g	Access Point
welcome@HTW	Open	None	-71	11	2462	802.11g	Access Point
vaipho	Open	None	-71	11	2462	802.11g	Independent
Gast@HTW	WPA2/PSK	AES-CCMP	-73	11	2462	802.11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-82	1	2412	802.11g	Access Point

Fig. 9 Network instances

Fig. 10 NS-2 simulation

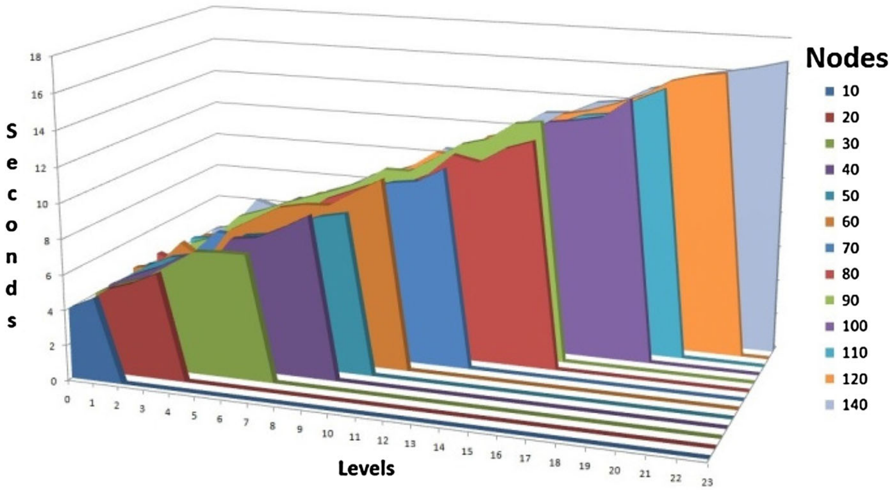

Fig. 11 Delay of sub-network merging

6.2 Large-scale simulation

The protocol to merge some sub-networks has a cost in time and number of communications. This protocol has been simulated in NS-2, and many data have been carried out for different sub-network sizes. Implementations with real devices allow us to obtain the average time that the devices need to restart their network interface and reconnect to another existing sub-network. The mid time for the proofs done with the HTC HD Mini device was 2.94 s, while the mid time for the HTC P3300 was 9.15 s to reconnect. The time required for the authentication process is not included here. Therefore, this data includes the mid time it takes to shut down the network interface, the waiting time, and the time that the device need to detect and reconnect to another vaipho sub-network. Obtained data with real devices were used for large-scale NS-2 simulations. For these simulations, the mobility model used for the VANET was a traffic jam. Figure 10 shows the implemented highway. It has three lanes each sense, a length of 1 km and a density about 0.1 vehicles per meter and lane. To distinguish the two sub-networks, these are in different colours while red is used for the node who detects the two existing sub-networks and starts the reconnecting protocol with the message to reset the network interface.

Twenty five simulations for each scenario ranging from 10 to 140 nodes per sub-network were simulated. Data from these simulations are shown in Fig. 11. The average time obtained to reconnect to another sub-network depends on the size, but the figure shows that it grows in a linear way. These simulations show the average time that nodes take to reconnect to the sub-network with less interference.

There exist different levels depending on the size and distance between nodes belonging to a sub-network. In such a figure, 'level' refers to the subset of nodes that are in same transmission range of one node, so the first node who detects that there exist multiple instances of the vaipho sub-network, sends a signal to start the reconnecting protocol, when this signal reaches the subset of nodes that are closest to it, they broadcast this information to the rest of nodes and later, they restart their network interfaces.

The following nodes that reach this information will be from another level, and so on. In this simulation, the number of nodes of the network is taken into account because the connectivity of the whole network depends on such a number. Other data, such as the amount of transferred data, could be also interesting, but the quality of the connections depending on these data is out of the scope of this work. The simulations show a logical result, the larger the size of the sub-networks, the longer the sub-network merging time. However, if the density of nodes is high and there are a few transmission levels, it takes about the same time regardless of the number of nodes that the sub-networks have (Fig. 11)

7 Security analysis

VANET topology does not allow guaranteeing complete security due to the fact that the wireless channel is open. There exist some types of difficulties or attacks that both unauthenticated and authenticated nodes can do by applying the weaknesses of a distributed and self-managed ad-hoc network like a VANET here analyzed. The two main types of attacks on these kind of networks that can be distinguished are the following: attacks from outside by non-authenticated nodes, and attacks from inside performed by authenticated nodes. In this paper, it is assumed that a robust authentication protocol makes impossible inside attacks.

At the moment to reconnect to a sub-network, there are some possibilities to create a blind spot. Nevertheless, in case of the sub-networks not join the main network, they would be isolated. Consequently, they would not be able to communicate each other. This problem is worse than the delay to send information, which may induce the whole network reconnection.

Attacks accomplished against the IEEE 802.11 protocol can be categorized into passive and active. The most relevant attacks can be categorized like: passive attacks as sniffing and passive traffic analysis, and active attacks like impersonation and DoS are significant. The majority of these attacks can be prevented with a strong authentication protocol.

DoS attacks can be easily done by any malicious user by introducing noise in the channel. Thereby, devices will not be able to communicate each other in the appropriate transmission range. A feasible method to avoid this attack is the channel change since that modifies broadcast frequency. One suitable active inside attack in the scheme here proposed would be a malicious user whom creates a false *vaipho* instance in a different channel from the original *vaipho* sub-network. At this way, it forces that the other devices have to restart their wireless interfaces. Therefore, if the malicious user repeat the attack many times, the nodes belonging to this sub-network would restart it constantly. To prevent this attack, first a strong authentication scheme is necessary. Then, a timestamp can be defined so that the devices have to wait for some time before reconnecting. This solution is not perfect, but it allows the proper operation of the network.

8 Conclusion

Two practical solutions for the problem that arises when merging several wireless networks formed by mobile devices equipped with Wi-Fi by using the IEEE 802.11xx protocol have been here proposed. A usual drawback occurs when sub-networks emerge on different channels so that some nodes can not see other nodes. The best solution would be for the smaller sub-networks to join the largest sub-network, but there is no possibility that the nodes know the number of devices that are in other sub-networks. This paper first presents a simple generic deterministic solution based solely on the number of nodes in a known sub-network. Then, it provides the outline of an approach based on fuzzy logic to estimate the time a node has to wait before checking if the problem is solved or not, so that in this case its network interface is reset. This method is based on both the number of nodes in its sub-network, and data on potential interferences with other sub-networks. Both proposals have been implemented in a new tool developed to create a VANET by using only mobile devices. The performance evaluation of implementations in real devices and large-scale NS2 simulations is promising as it shows how in a few seconds, all nodes of all sub-networks merge into a single network.

References

1. Arani, F., Smietana, R., Honary, B.: Real-time channel estimation using Fuzzy logic, IEEE International Symposium on Information Theory, pp. 289, (1995)
2. Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J.: Self-organizing life cycle of mobile ad-hoc networks. Secur. Commun. Netw. **5**(10), 1147–1158 (2012)
3. Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J.: VAiPho: VANET application for mobile Phones to avoid traffic jams. University of La Laguna, Spain, International Patent Number PCT/ES2011/000220, http://www.vaipho.com, (2010)
4. Gruenwald, L., Banik, S.M., Lau, C.N.: Managing real-time database transactions in mobile ad-hoc networks. Distrib. Parallel Database **22**(1), 27–54 (2007)
5. Hauser, J.P., Baker, D.J.: Dynamic backbone subnets - DBS/802.11 access point-to-access point routing for 802.11 infrastructure networks, IEEE 802 Plenary Meeting, (2002)
6. Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C: Cooperation enforcement schemes in vehicular ad-hoc networks. Computer Aided Systems Theory-EUROCAST, pp. 429–436 (2009)
7. Kumar, H., Singla, R.K., Malhotra, S.: Issues & trends in auto configuration of IP address in MANET. International Conference on Computer, Communication and Control, pp. 15–17, (2008)
8. La Kloul, L., Valois, F.: Investigating unfairness scenarios in MANET using 802.11b, ACM International Workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pp. 1–8, (2005)
9. Lin, B.Y., Chen, C.H., Lo, C.C.: A novel speed estimation method using location service events based on fingerprint positioning. Adv. Sci. Lett. **4**, 11–12 (2011)
10. Liu, M., Lai, T.H., Liu, M.T.: Is clock synchronization essential for power management in IEEE 802.11-based mobile ad-hoc networks, IEEE International Conference on Mobile Ad-hoc and Sensor Systems, (2005)
11. Luo, J., Li, W., Liu, B.: Distributed intelligent network management model for the large-scale computer network. Comput. Support. Coop. Work Des. **3865**, 313–323 (2006)
12. Mantoro, T., Ayu, M., Raman, S., Latiff, N.: Particle filter approach for tracking indoor user location using IEEE 802.11 signals. Adv. Sci. Lett. **9**(1), 86–91 (2012)
13. Nawaz, R., Sun, S.: Channel estimation in 802.11G in the presence of bluetooth interference, European Signal Processing Conference, (2008)

14. Ramachandran, K., Belding, E., Almeroth, K., Buddhikot, M.M.: Interference aware channel assignment in multi-radio wireless mesh networks. IEEE INFOCOM **6**, 1–12 (2006)
15. Skalli, H., Ghosh, S., Das, S.K., Lenzini, L., Conti, M.: Channel assignment strategies for multiradio wireless mesh networks: issues and solutions. IEEE Commun. Mag. **45**(11), 86–95 (2007)
16. Tanaka, H., Masugata, O., Ohta, D., Hasegawa, A., Davis, P.: Fast, self-adaptive timing synchronisation algorithm for 802.11 MANET. Electron. Lett. **42**(16), 932–933 (2006)
17. Wilson, H., Walr, J.: A multi-channel MAC proposal for ad-hoc wireless networks, IEEE Wireless Communications and Networking Conference, (2007)
18. Xing, Z., Gruenwald, L.: Managing concurrent execution of transactions in mobile ad-hoc network database systems: an energy-efficient approach. Distribut. Parallel Databases **31**(2), 183–230 (2013)
19. Yang, K., Wang, X.: Network planning for multi-radio cognitive wireless networks, European Signal Processing Conference, (2006)