Peer-to-Peer Netw. Appl. (2015) 8:664–673
DOI 10.1007/s12083-014-0313-z

Merging sub-networks in VANETs by using the IEEE 802.11xx protocols

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

Received: 31 August 2012 / Accepted: 15 September 2014 / Published online: 15 October 2014

Abstract Nowadays most mobile ad-hoc networks are based on the IEEE 802.11 standards for wireless technology under the Wi-Fi brand. The structure of such protocols produces several problems when distinct sub-networks are to be merged. The main troubles that can cause inability to communicate are IP duplication and the existence of sub-networks on different channels. A practical solution to these issues has been fully implemented in a new tool developed to create a vehicular ad-hoc network by using only smartphones. This paper proposes both a simple and deterministic algorithm, and a more complex procedure that considers interferences between wireless channels under a fuzzy logic approach. Both from the performance and security points of view, encouraging results were extracted from the analysis of large scale simulations based on data obtained through executions on smartphones.

Keywords Sub-networks · VANETs · IEEE 802.11

1 Introduction

A Vehicular Ad-hoc NETwork (VANET) is a specific version of a Mobile Ad-hoc NETwork (MANET) where the mobile nodes are vehicles that can communicate with each other in order to increase comfort and safety of every day road travel. The classical VANET definition suggests connections IEEE 802.11p specially designed for vehicle communications. In such a proposal, On Board Units (OBUs), which are installed in vehicles, and Road Site Units (RSUs), which are installed in roads, are assumed. However, these devices are not yet available. For this reason in this research we propose the IEEE802.11b/g communication protocol, that is present in most of the current smartphones. The IEEE 802.11 protocol is used in most existing MANETs to deploy Wi-Fi connection, which enables short-to-medium-range wireless communication capability (up to 300 m). In these 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 wireless encouraging conclusions network in real environments can cause trouble mainly due to the spontaneous formation of sub-networks. The two most relevant problems 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 most usual problem is the first one because the choice of the wireless channel is random. Furthermore, the IP duplication problem has a simple solution connected to the proposed solution to the merging problem.

The usual broadcasting channels in the IEEE 802.11xx protocol are a, b, g and n. It ranges between channels 1–11 in America, 1–13 in Europe, and 1–14 in Japan, and in every case some intersection exists among the bandwidth of the channels. Figure 1 shows the channels that are available in Europe to create wireless networks, highlighting the maximum number of channels that can be used with no interference. In particular five is the maximum number of simultaneous wireless networks with no interference.

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

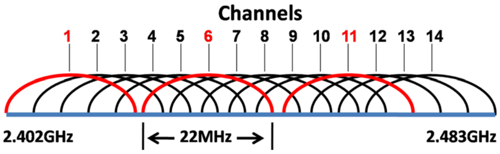


Fig. 1 European channels

All the versions of the IEEE 802.11xx protocol are compatible with each other, so the user does not require anything apart from its integrated Wi-Fi adapter. However, too much saturation in one channel can cause packet collisions in the data transfer, what corresponds to a lower transfer rate and/or speed. The ideal solution consists in that existing sub-networks share a channel with no IP address conflict so that they can merge. However, this situation does not appear frequently.

This paper describes practical solutions for both problems mentioned above, which furthermore have been implemented through a new application to generate a VANET with only Wi-Fi smartphones.

This paper is organized as follows. The first section briefly describes some related papers while Section 2 defines the basis of our proposal. Sections 4 and 5 introduce a generic deterministic proposal, and a fuzzy logic based approach respectively. Sections 6 and 7 include respectively a performance and a security analysis. Section 8 gives conclusions and open problems.

2 Related work

Recently, much research has been developed in the fields of MANETs and VANETs. [3] presents a self-organizing life cycle management for MANETs and [9] deals with the topic of VANETs and the building of real-time traffic information system. A strategy of service execution and service recovery in MANETs is proposed in [19]. While most studies on these topics are rather theoretical, some include simulations using tools such as NS2 [2] or OPNET [5], and a few show implementations in real environments [6]. In this work, two different methods are proposed and checked by combining software simulations with data obtained from implementations in real devices.

Different problems of the IEEE 802.11 protocol are addressed in several papers. [7] includes a performance analysis of the 802.11b protocol using analytical modelling. [10] studies the relationship between clock synchronization and power-saving in IEEE 802.11-like MANETs. [12] discusses the accuracy of IEEE 802.11 signals in indoor location. [13] proposes and simulates an algorithm to mitigate the Bluetooth interference with the channel estimation stage in IEEE 802.11 g. [11] focuses on the quality of service techniques over IEEE 802.11e networks. All the above proposals deal with problems of the IEEE 802.11, but none of them follows the approach of this paper.

The problem that appears when an IP address conflict exists during the sub-network merging is addressed by the authors of [8]. [17] tries to solve different problems regarding channels in ad-hoc networks, where multiple channels are used simultaneously. [1] shows a way to estimate and avoid noise in the channel by using fuzzy logic. [18] proposes a framework for multi-radio multi-channel cognitive wireless networks to maximize a network utility function based on data routing, resource allocation and scheduling. [15] studies the assignment of non-overlapping channels in wireless mesh networks by minimizing interference while improving the network capacity and maintaining the connectivity of the network. The same problem in mesh networks is addressed in [14], which proposes a dynamic centralized interference-aware algorithm aimed at minimizing interference.

One of the closest works to this one is [16]. It studies multi-hop ad hoc networks using conventional IEEE 802.11 where long transient resynchronization states are often generated. The authors propose a modification of the resynchronization that reduces times and energy consumption. However, its approach is completely different, and it does not include any implementation in real devices, what is one of the main aspects of the present paper.

3 Basis of the proposal

The paper [4] describes a new application, called Vnetwork, to create a VANET using smartphones playing the role of On-Board Units (OBUs) inside vehicles. Different types of events, such as traffic jams, are automatically detected in real time, and warnings about them are sent to other smartphones through Wi-Fi, what provides drivers with an augmented reality on road conditions.

Such a tool requires that each smartphone acts at the same time as a client and as a server, and connects to the same wireless network. This tool is especially useful for urban environments where limit speed is low. It also works in vehicles on highways in the same works in vehicles on highways in the same direction and at junctions. However, unfortunately, it does not work for vehicles traveling in opposite directions at high speed.

In the VANETs deployed with Vnetwork, a problem can happen if there are two nodes of the same network on different channels. This issue requires sub-network merging. The ideal solution would imply that the largest sub-network absorbs the smallest one, but nodes do not know the size of other sub-networks. Once connected both sub-networks, the number of communications on the selected channel grows, but there are proposals to reduce the number of communications overhead [2] that could be after the merging in the channel.

The first proposal included in this work is a basic deterministic solution to compute the minimum time that any device of a sub-network requires to reset and to connect with other sub-network, depending on the size of its sub-network. Then, fuzzy logic rules are used to interpret interferences between channels in terms of sizes of neighbour sub-networks in order to allow that larger sub-networks absorb 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 approach

This paper proposes a solution to the sub-network merging problem based on resetting the wireless interface of the devices belonging to all but one wireless sub-networks.

The optimal solution would imply resetting sub-networks with the lowest number of 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 simply resets its network interface, what solves the problem. However, the problem is more difficult when the sub-networks are on different channels.

The simplest solution is based on choosing the sub-networks in the most appropriate channel to minimize the interference created by the networks on adjacent channels (see Algorithm 1). Algorithms are written in C# as the implementation is for Windows Mobile. Each device running Vnetwork regularly checks for existing Vnetwork sub-networks.

Algorithm 1 Auto detect the best sub-network to operate

```
//detect that several Vnetworks exist
int[] channelDifference = new int[nNetworks]; // interference (dB)
for (int i = 0; i < VnetworksInstances; i++) //initialize
    channelDifference[i] = 50;
for (int i = 0; i < nNetworks; i++)
    for (int j = i+1; j < nNetworks; j++)
        if ((Math.abs(channel[i] - channel[j])) < channelDifference[i])
            channelDifference[i] = Math.abs(channel[i] - channel[j]);
    //the channel with the 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();
            return true;
```

If there are two instances of the network called Vnetwork exist, all the devices belonging to one of the sub-networks have to reset their wireless interfaces. The first node that detects this situation sends a warning message to the remaining nodes so that they restart its wireless interface. Thus, upon receipt, the nodes broadcast the message, turn off their network interface for 1 s and reactivate it. Then, since a Vnetwork network already exists when they restart, the nodes will connect to this network and the problem will be solved. The nodes will remain authenticated by the nodes of the previous sub-network because authentication data do not vary with the change of sub-network, which has the same name, Vnetwork.

This solution may have the problem of mutual reconnection. If two devices on different channels detect the existence of another sub-network at the same time, it could restart both sub-networks. However, in this case, the creation of two new sub-networks during the reconnection of the devices is unlikely.

When different sub-networks are neighbours, each device of each sub-network can determine the existence of other sub-networks. To enable the deployment of VANETs through Vnetwork, each device has to check whether other Vnetwork sub-networks exist in its transmission range. If different sub-networks exist with the same name because they were created independently on different channels, these sub-networks must be merged. Each member of one of the sub-networks that discovers the problem must choose a new channel. In particular, the devices in the sub-network with the largest channel number will restart their network interfaces because the new channel of the merged sub-networks will be the smallest number of all previous channels. However, such a way to choose the surviving sub-network is not the best one because it does not depend on the number of nodes in such a sub-network, or the interferences between channels used by other wireless networks. Each device can know only the number of nodes that are in its sub-network, the channel where it is, in which channels there are other instances of the Vnetwork network, and the channels that are being used by other networks that can interfere with its Vnetwork sub-network. In fact, the detected interference in the channels can be used to determine which sub-network must be restarted.

Figure 2 shows a usual example of a VANET in an urban scenario where two sub-networks of vehicles denoted A and B enter the same transmission range and find out that two Vnetwork instances were formed on different channels while there are 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 analyse possible interferences with other wireless networks. In Fig. 2 the sub-network A is on channel 1, B is on channel 6, and two other wireless networks are on channels 2 and 4. Therefore, the channel of A, which is 1, is at distance 1 from the channel of the nearest network, which is channel 2; while the channel of B, which is 6, is at distance 2 from the channel of the nearest network, which is 4. Thus, there should be concluded that the channel of the sub-network A has more possibilities of interference, and consequently those devices should restart to merge with sub-network B. However, the best solution should consist in resetting the sub-networks with fewer devices. Thus, in the next section a new approach based on fuzzy logic is proposed as a closer to the optimal solution.

The second challenge we have faced is the problem of IP duplication. However, this problem is easily solved if the first node that detects the IP duplication automatically resets its device.

5 Fuzzy logic based approach

Fewer packets are generated and lost during a merging process if the sub-network whose devices restart its network interface is the one with fewer nodes, as can be seen in Fig. 3. Data shown in this figure were obtained through simulation executed with data got from real experiments whose parameters are described in Section 6. Figure 3 also shows that the number of lost packets grows with the number of nodes but this number is very small and can be ignored. This happens because vehicles are evenly spaced, what makes that the signal is not saturated. From this starting point, we propose a heuristic method that, without knowing the number of nodes in other sub-networks, allows that the sub-network with fewer devices is the sub-network whose nodes restart their wireless interfaces. In addition, another key factor is the interference with the channels of other existing sub-networks.

In the following, a fuzzy logic approach is used to compute the time that each node has to wait before checking whether the problem remains unsolved, so that in that case, it begins the merging process. For the estimation of such a 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 goals regarding the waiting time computation are that the times are minimized and that the nodes belonging to the chosen sub-network do not restart their interface because this would imply that, in fact, all the nodes would restart their interfaces.

A(x) represents the time that each node waits before checking whether it has to restart and join another sub-network is first expressed in terms of the number x of nodes in its sub-network. For the fuzzy logic approach, the times shown in Fig. 4 were computed with real devices. This time 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, or represented by the logarithmic Eq. 2.

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

$$A(x) = 6.2454ln(x)–0.6499 \pm Sr \ if \ x > 0 \tag{2}$$

Both expressions have been used as starting points to estimate the average time before checking whether multiple Vnetwork instances exist. In particular, the use of the linear expression by removing the initial constant time *3.63* is proposed. In this way, a node detecting multiple instances of Vnetwork network waits for a time depending on the number of nodes in its sub-network *W(x)= 0.107x*. For instance, a node detecting multiple instances of Vnetwork network such that its sub-network has 2 nodes, will wait for 0.214 s before rechecking whether there are more than one Vnetwork instance, and another node detecting the same but belonging to a 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.

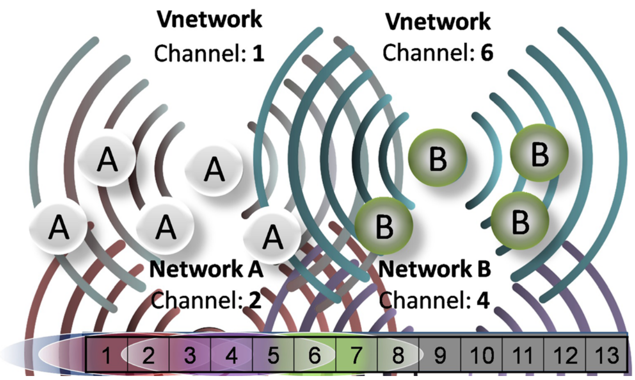


Fig. 2 VANET example

Apart from taking into account the number of nodes in each sub-network, for sub-networks with similar size, the interference between channels should be also considered so that the sub-network with less interference prevails over the others. A sub-network operating on a channel is said to have no interference if no other sub-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. 3.

$$dBm = 10\log\frac{P}{1mW} \tag{3}$$

By using this metric, the interference *I(c)* that a network using a channel *c* has can be estimated through Eq. 4, where *M* denotes the number of sub-networks using channel *c*; *N*, *P*, *Q* and *R* denote respectively the numbers of sub-networks within 1, 2, 3, or 4 channels of distance from channel *c*, and *a* is got from the expression $\frac{2a}{dBm_{high}} = 1$ where *dBmhigh* corresponds to the channel with the highest interference.

$$I(c) = \sum_{i=1}^{M} \frac{-a}{dBm_i} + \sum_{j=1}^{N} \frac{4(-a)}{5dBm_j} + \sum_{k=1}^{P} \frac{3(-a)}{5dBm_k} + \sum_{l=1}^{Q} \frac{2(-a)}{5dBm_l} + \sum_{m=1}^{R} \frac{-a}{5dBm_m} \tag{4}$$

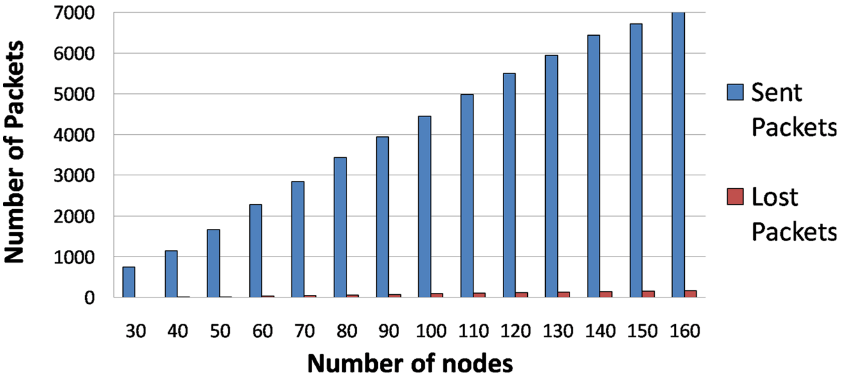


Fig. 3 Generated and lost packets

The above expression results from that if two sub-networks operate 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 place where vehicles are because if they are in a location with many wireless networks, the possibility of having more than two sub-networks using the same channel is higher.

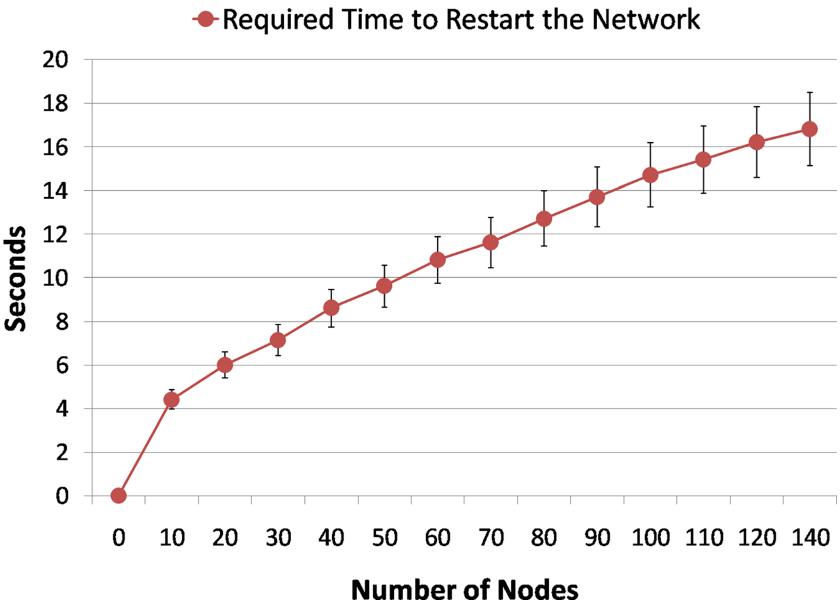
At this point, the use of fuzzy logic for the application of the interference between channels of sub-networks is proposed to estimate the waiting times for checking and resetting. From real data got from the experiments we have concluded that the expression defining a degree of interference *B* for a Vnetwork instance that depends on the interference *d* (resulting from *I(c)* in eq. 4) of the own sub-network can be estimated according to the equations below, where Eq. 5 corresponds to the case without interference with other wireless networks, and Eq. 6 corresponds to when interference with other wireless networks exists.

$$B(d) = \begin{cases} \nexists & if\, d > 0 \\ 1 + 0.11d & if\, d \in (0, -90) \\ 0 & if < -90 \end{cases} \tag{5}$$

$$B(d) = \begin{cases} \nexists & if \;\; d \;\; > \;\; 0 \\ 0 & if \;\; d \in (0, -90) \\ -0,005(d+10) & if\, d \in (-10, -210) \\ 1 & if \;\; d \;\; < \;\; -210 \end{cases} \tag{6}$$

In order to estimate the waiting time for the sub-network merging we consider the degree of interference between two instances of Vnetwork network. Figure 5 shows an estimation of such a degree depending both on the power of the network and on the existence of interferences with other networks. From it, we get the design of fuzzy control rules, denoted by Rules 1 and 2, which depend on the interferences of the network of the node

Fig. 4 Time for sub-network merging



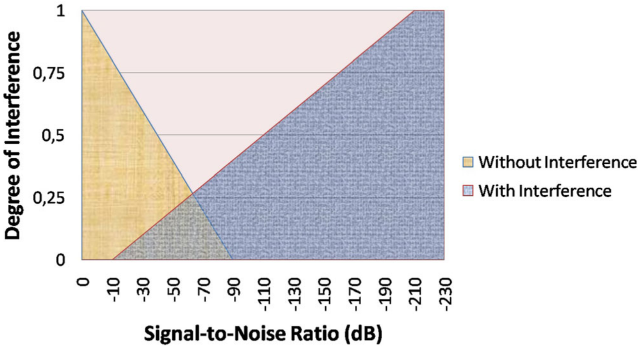


Fig. 5 Fuzzy logic approach

performing the test, denoted by Vnetwork(α); and of the other sub-networks, denoted by Vnetwork(β(l)) where *l≥1*.

F(x) denotes the time that the *x* nodes of the sub-network Vnetwork(α) wait before checking again whether other Vnetwork instances exist. If a different Vnetwork instance exists after this waiting time, the node detecting it, starts the merging protocol. Otherwise, it does not restart its interface. Rule 2 is always tested after Rule 1 to determine the time that is added or subtracted to the waiting time *F(x)*.

```
Rule 1 Own instance of the $Vnetwork$ network
if $(I(Vnetwork(\alpha)) > -10)$ then //With interference
    $F(x) = W(x) + \frac{DI(Vnetwork(\alpha))}{2}$
else //Without interference
    $F(x) = W(x) - \frac{DI(Vnetwork(\alpha))}{2}$
```

```
Rule 2 Other instances of the $Vnetwork$ network
if $(I(Vnetwork(\beta(l))) < -90)$ then //Without interference
    $F(x) = F(x) - \frac{DI(Vnetwork(\beta(l)))}{2}$
else //With interference
    $F(x) = F(x) + \frac{DI(Vnetwork(\beta(l)))}{2}$
```

Each sub-network Vnetwork(α) with *x* nodes estimates its own waiting time *F(x)* by considering not only its own data but also another sub-network Vnetwork(β(l)) data. Figure 6 shows the waiting time a node waits before rechecking Vnetwork interfaces, which depends on the number of nodes of the corresponding sub-network and on the interferences in the channels of Vnetwork(α) and Vnetwork(β(l)).

6 Performance analysis

In order to check the performance of the proposal, the best option would be to check it with many real devices, but that is not feasible, so the chosen alternative has been to implement the scheme in a few real devices, in order to obtain real data of reconnection and merging, so that we can use those real data in a NS2 simulation.

The real device implementation has been based on mobile phones with Windows Mobile 5 and 6 (see Table 1), and Microsoft Visual Studio 2008.

A screenshot of two emulators using the same Vnetwork sub-network while there is another Vnetwork sub-network in another channel is shown in Fig. 7. The device on the right is the first one to detect the presence of multiple instances of the network Vnetwork. After checking that there are other Vnetwork instances, it starts the merging protocol, so that the node detecting the other sub-networks, broadcasts a warning message to the other nodes of its sub-network to make they restart their network interfaces. After that, every node turns off its interface for some time, reactivates it, and connects to the existent Vnetwork network.

Figure 8 shows several wireless networks in the transmission range. It can be seen two instances of Vnetwork network on channels 1 and 11, and also other networks there. Thus, merging is required.

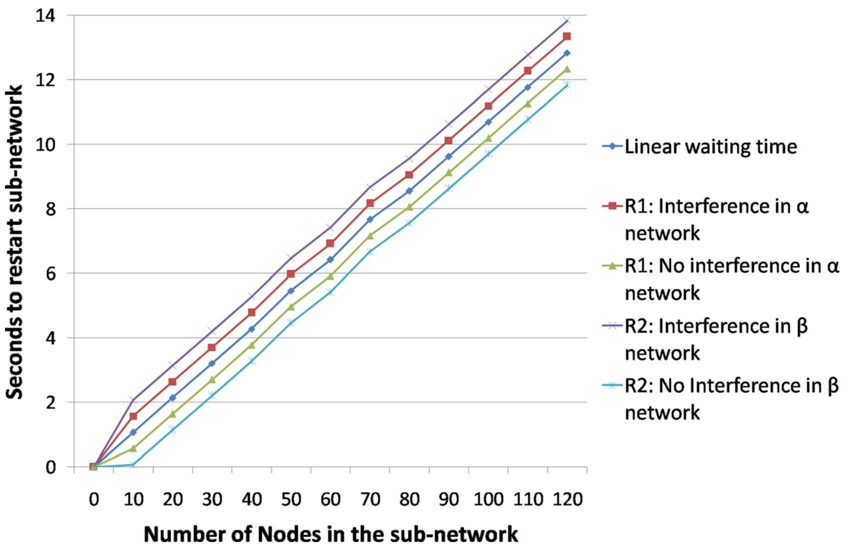


Fig. 6 Waiting time

Table 1 Used smartphones

Phone name	Platform	CPU Speed	RAM	ROM	Battery capacity
HTC HD Mini	WM 6.5	600 MHz	384 MB	512 MB	1,200 mAH
HTC P3300	WM 5.0	201 MHz	64 MB	128 MB	1,250 mAH
HTC Touch2	WM 6.5	528 MHz	256 MB	512 MB	1,100 mAH
hp IPAQ 614C	WM 6.0	520 MHz	128 MB	256 MB	1,590 mAH

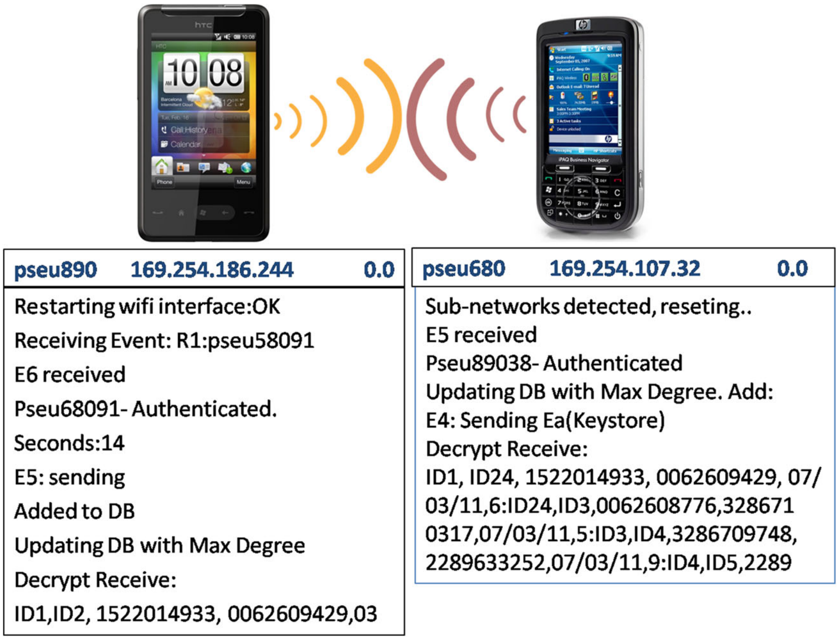
In order to check the average time to merge any sub-network into another sub-network, several NS2 simulations were carried out for different sub-network sizes. The average time that the devices take to restart their network interface and connect to other network instance was taken from the implementations with real devices. The best case was 2.94 s for the HTC HD Mini, while the worst case was 9.15 for the HTC P3300. Times required for authentication were not included here. Considered data only includes the time it takes to shut down the network interface plus the time for the waiting process, and the time to reconnect the device to the existing Vnetwork network. After the executions with real devices, the obtained data were used for a large-scale NS2 simulation. Figure 9 shows a simulation of a highway with three lanes, length of 1 km and a density about 0.1 vehicles per meter and lane in the traffic jam, where the two sub-networks are in different colours and red is used for the node that detects the two network instances and broadcasts the message to reset the interface.

Figure 10 shows the numerical results obtained from the NS2 simulations. These simulations were performed with sub-networks of sizes between 10 and 140 nodes and 25 simulations for each scenario. We obtained the average times that nodes take to reconnect to the network with less interference. Different levels were distinguished depending on the size of the sub-network. In such a figure, we use the word 'level' to refer to the subset of nodes that are in same transmission range of the transmitter node, so the first node detecting that there are multiple instances of Vnetwork network, sends a warning message, when the warning reaches the subset of nodes that are closest to it, they broadcast this information and then, restart their network interfaces.

Nodes that reach this information will be from another level, and so on. Figure 10 shows the average time it takes for the nodes in each transmission level to reconnect to the other Vnetwork sub-network. The simulations show a natural result, the larger the size of the sub-networks, the longer the sub-network merging time. However, if nodes are divided into transmission levels, it can be seen that it takes about the same time regardless of the number of nodes that the sub-networks have.

7 Security evaluation

The topology of VANETs does not allow guaranteeing absolute security mainly because the wireless channel is open. There are several types of problems or attacks that both unauthenticated and authenticated nodes can perform by

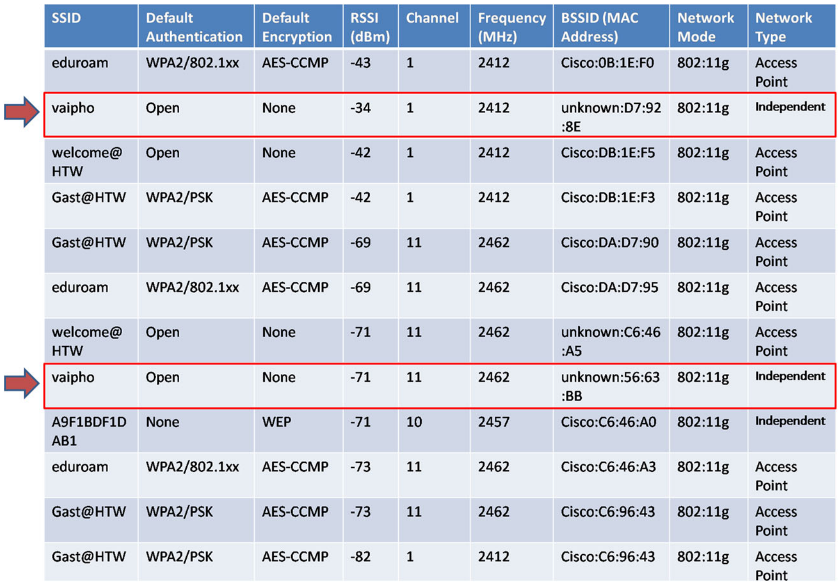


```
pseu890     169.254.186.244          0.0
Restarting wifi interface:OK
Receiving Event: R1:pseu58091
E6 received
Pseu68091- Authenticated.
Seconds:14
E5: sending
Added to DB
Updating DB with Max Degree
Decrypt Receive:
ID1,ID2, 1522014933, 0062609429,03
```

```
pseu680     169.254.107.32          0.0
Sub-networks detected, reseting..
E5 received
Pseu89038- Authenticated
Updating DB with Max Degree. Add:
E4: Sending Ea(Keystore)
Decrypt Receive:
ID1, ID24, 1522014933, 0062609429, 07/
03/11,6:ID24,ID3,0062608776,328671
0317,07/03/11,5:ID3,ID4,3286709748,
2289633252,07/03/11,9:ID4,ID5,2289
```

Fig. 7 Executions on smartphones

Fig. 8 Network examples



SSID	Default Authentication	Default Encryption	RSSI (dBm)	Channel	Frequency (MHz)	BSSID (MAC Address)	Network Mode	Network Type
eduroam	WPA2/802.1xx	AES-CCMP	-43	1	2412	Cisco:0B:1E:F0	802:11g	Access Point
vaipho	Open	None	-34	1	2412	unknown:D7:92 :8E	802:11g	Independent
welcome@ HTW	Open	None	-42	1	2412	Cisco:DB:1E:F5	802:11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-42	1	2412	Cisco:DB:1E:F3	802:11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-69	11	2462	Cisco:DA:D7:90	802:11g	Access Point
eduroam	WPA2/802.1xx	AES-CCMP	-69	11	2462	Cisco:DA:D7:95	802:11g	Access Point
welcome@ HTW	Open	None	-71	11	2462	unknown:C6:46 :A5	802:11g	Access Point
vaipho	Open	None	-71	11	2462	unknown:56:63 :BB	802:11g	Independent
A9F1BDF1D AB1	None	WEP	-71	10	2457	Cisco:C6:46:A0	802:11g	Independent
eduroam	WPA2/802.1xx	AES-CCMP	-73	11	2462	Cisco:C6:46:A3	802:11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-73	11	2462	Cisco:C6:96:43	802:11g	Access Point
Gast@HTW	WPA2/PSK	AES-CCMP	-82	1	2412	Cisco:C6:96:43	802:11g	Access Point

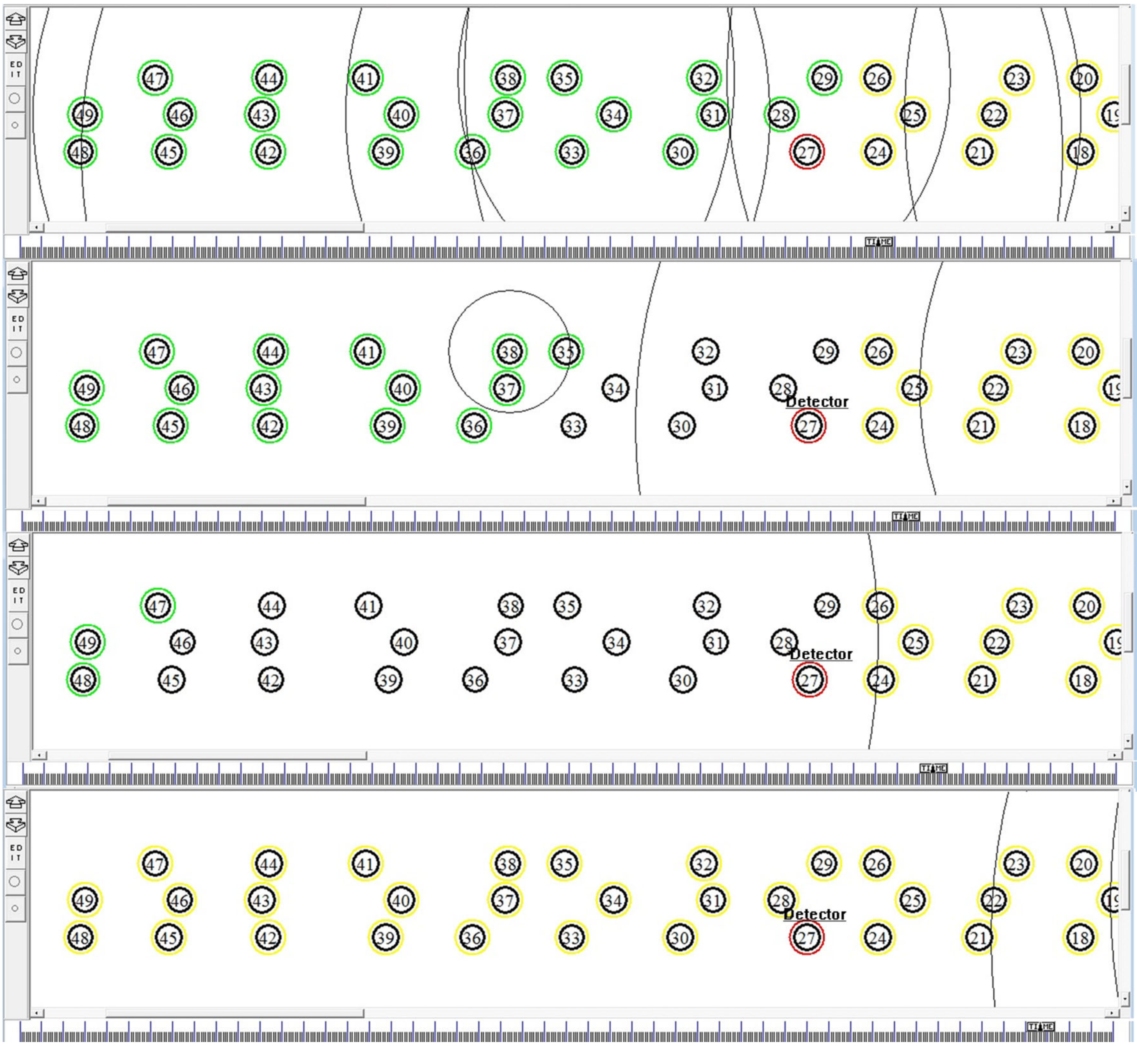


Fig. 9 Simulation

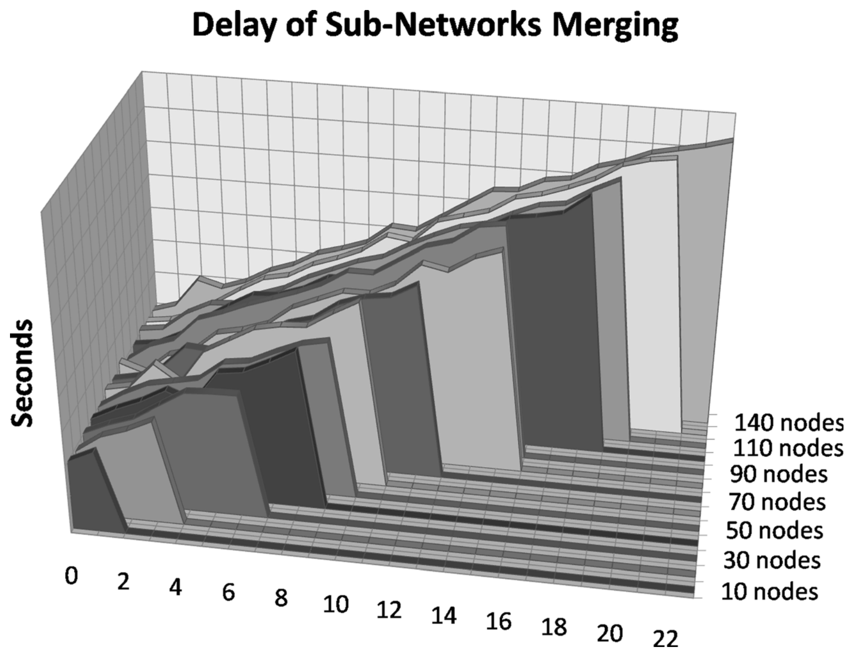


Fig. 10 Merging delay

using the vulnerabilities of a distributed and self-managed wireless network like the VANET here analysed. Two types of attacks on wireless networks can be distinguished: attacks from non-authenticated nodes (outside attacks), and those produced by authenticated nodes (inside attacks). This paper assumes that a strong authentication protocol prevents inside attacks.

When reconnecting to a subnetwork, it is possible to generate a blind spot. However, if the subnetworks were not reconnected, they would be isolated. Therefore, they would unable to exchange any event. This situation is worse than the delay, in sending information, which could cause the reconnection.

Attacks carried out against IEEE 802.11 can be classified into passive and active. Among the most important attacks, passive attacks such as sniffing and passive traffic analysis, and active attacks such as impersonation and DoS are remarkable. Most attacks can be avoided with a good authentication protocol.

Any malicious user who can introduce noise in the channel can perform DoS attacks. In this way the devices cannot communicate with each other in the corresponding transmission range. A possible solution to this attack is the channel change because that produces the change in the broadcast frequency. The proposed scheme can suffer an active inside attack consisting in that a malicious user could create a false Vnetwork instance in a different channel from the original Vnetwork sub-network to force that the other devices restart their wireless interfaces. So, if the malicious user does this once and again continuously, the nodes would be reconnecting all the time. This attack can be prevented by using a strong authentication scheme and a timestamp. In this way, the devices have to wait for some time before reconnecting. Although this solution is not perfect, the proper operation of the network can be made possible by applying it.

8 Conclusion

This paper includes the proposal of two practical solutions for the problem that appears when it is necessary to merge several wireless networks formed by smartphones equipped with IEEE 802.11xx Wi-Fi. A usual problem arises when sub-networks are created on different channels so that some nodes are not visible to other nodes. The best solution would consist in that the smaller sub-networks join the largest sub-network, but there is no possibility for nodes to know the number of devices that integrate other sub-networks. Thus, this paper first proposes a generic deterministic solution based only on the number of nodes in a sub-network. Then, it outlines a fuzzy logic approach to estimate the time a node has to wait before checking whether the problem has been solved or not, so that in this case it restarts its network interface. This method is based on the number of nodes in its sub-network, and on some data about possible interferences from other sub-networks. Both proposals have been implemented in a new tool developed to create a VANET by using only mobile devices. The performance of the implementations carried out with real devices and software simulation produced promising results because during the execution, all the nodes of all the sub-networks were able to merge into a single network just in a few seconds.

References

1. F. Arani, R. Smietana, B. Honary, Real-Time Channel Estimation Based on Fuzzy Logic, IEE Colloquium on Frequency Selection and Management Techniques for HF Communications, p. 10, 1996.
2. C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Using groups to reduce communication overhead in VANETs, the second international conference on advances in P2P systems, 2010.
3. C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Self-organizing Life Cycle Management of Mobile Ad hoc Networks. FTRA International Symposium on Advances in Cryptography, Security and Applications for Future Computing ACSA, 2011.
4. P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, VAiPho (VANET in Phones), http://www.vaipho.com, Patent P201000865, University of La Laguna, Spain, 2011.
5. A. Ghosh, A. Lasebae, E. Ever, Performance Evaluation of Wireless IEEE 802.11(b) used for Ad-Hoc Networks in an ELearning Classroom Network, Kaspersky Lab IT Security Conference for the Next Generation, 2009.
6. J.P. Hauser, D.J. Baker, Mobility and routing protocols for 802.11 extended service sets. IEEE Military Communications Conference 2, pp. 1036–1041, 2003.
7. L. Kloul, F. Valois, Investigating unfairness scenarios in MANET using 802.11b, ACM International Workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, New York, 2005.
8. H. Kumar, R.K. Singla, Architecture for address auto-configuration in MANET based on extended prime number address allocation, WSEAS Transactions on Computers 8(3), pp. 549–558, 2009.
9. B.Y. Lin, C.H. Chen, C.C. Lo, A novel speed estimation method using location service events based on fingerprint positioning. Adv Sci Lett 4(11–12), pp. 3735–3739, 2011.
10. M. Liu, T.H. Lai, M.T. Liu, 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. H. Luo, M-L. Shyu, Quality of service provision in mobile multimedia - a survey, Human-centric Computing and Information Sciences 1(5), 2011
12. T. Mantoro, M. Ayu, S. Raman, N. Latiff, Particle filter approach for tracking indoor user location using IEEE 802.11 signals. Advanced Science Letters, 9(1), pp. 86–91, 2012.
13. R. Nawaz, S. Sun, Bluetooth Interference mitigation in 802.11 g. IEEE International Conference on Communications, pp. 930-935, 2008.
14. K. Ramachandran, E. Belding, K. Almeroth, M.M. Buddhikot, Interference Aware Channel Assignment in Multi-Radio Wireless Mesh Networks, IEEE INFOCOM, 2006.
15. H. Skalli, S. Ghosh, S.K. Das, L. Lenzini, M. Conti, Channel assignment strategies for Multiradio wireless mesh networks: Issues and solutions. IEEE Commun Mag 45(11), pp. 86–95, 2007.
16. H. Tanaka, O. Masugata, D. Ohta, A. Hasegawa, P. Davis, Fast, self-adaptive timing synchronisation algorithm for 802.11 MANET. Electron Lett 42(16), pp. 932–934, 2006.
17. J. Walrand, Comparison of Multichannel MAC Protocols. IEEE Transactions on Mobile Computing 7(1), 2008.
18. K. Yang, X. Wang, Cross-layer network planning for multi-radio multi-channel cognitive wireless networks. IEEE Transactions on Communications 56(10), pp. 1705–1714, 2008.
19. X. Zhou, Y. Ge, X. Chen, Y. Jing, W. Sun, A distributed cache based reliable service execution and recovery approach in MANETs. J Converg 3(1), pp. 5–12, 2012.

Cándido Caballero-Gil received his B.Sc. degree in Computer Science at the University of Las Palmas de Gran Canaria (Spain) and his Ph.D. degree in Computer Engineering at the University of La Laguna (Spain). Since 2012 he has been working under a research contract at the University of La Laguna. He is involved in several research projects and publications related to wireless security within the CryptULL research group devoted to the development of projects on cryptology. He has authored some refereed conference papers and journal articles.

Pino Caballero-Gil graduated with a B.Sc. and a Ph.D. in Mathematics from the University of La Laguna (Spain) in 1990 and 1995, respectively. Since 1990 she has been working at the University of La Laguna where she is now full professor of Computer Science and Artificial Intelligence. Her area of expertise includes security of wireless networks, cryptanalysis and cryptographic protocols. She leads the CryptULL research group devoted to the development of projects on Cryptology. She has authored many refereed conference papers, journal articles and books.

Jezabel Molina-Gil received her B.Sc. degree in Computer Science at the University of Las Palmas de Gran Canaria (Spain) and her Ph.D. degree in Computer Engineering at the University of La Laguna (Spain). Since 2012 she has been working under a research contract at the University of La Laguna. She is involved in several research projects and publications related to wireless security within the CryptULL research group devoted to the development of projects on cryptology. She has authored some refereed conference papers and journal articles.