Computer Communications 000 (2015) 1–12

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels☆

Yalew Zelalem Jembre, Young-June Choi*

Department of Computer Engineering, Ajou University, Suwon 443-749, South Korea

ARTICLE INFO

Article history:
Received 30 January 2015
Revised 2 September 2015
Accepted 3 September 2015
Available online xxx

Keywords:
Ad-hoc networks
Channel assignment
Multiple interfaces
Jamming

ABSTRACT

The capability of accessing multiple channels through multiple interfaces improve network capacity and is desirable for future Mobile Ad-Hoc Networks (MANETs). However, due to the presence of jammers as well as mobility and ad-hoc features, MANETs require distributed and efficient resource management for channel assignment. To address the channel assignment problem, which is a non-deterministic polynomial-time hard (NP-hard) problem, we propose a heuristic algorithm called Channel Assignment and JAmmer Mitigation (CA-JAM). The CA-JAM algorithm assigns a distinct channel for every interface of one station, and then all stations exchange the assignment information through beacon frames on every individual interface. When one station receives a beacon, the station organizes the information into tables. Therefore, each station, distributively, uses the table to reduce the number of neighboring stations using the same channel to avoid interference which in turn improves the throughput. The tables are also used to learn the disconnected neighbors due to jamming so as to mitigate the effect of jamming and maintain connectivity. CA-JAM is fully distributed with no use of control channel or central entity; thus, it improves connectivity and reduces interference by balancing stations over the available channels while mitigating jamming effects from multi channel multi interface MANETs. We confirm that CA-JAM outperforms existing protocols using the OPNET simulator.

1. Introduction

Mobile ad-hoc networks (MANETs) provide flexibility and scalability to set up a network compared to infrastructure networks. Such networks can be used in military combat, disaster relief or large construction sites. All MANET stations with a single interface or radio[1] should belong to the same service set on a single channel to stay connected, even if the interface can switch between channels [1]. As the number of MANET stations increases, interference and collision among them increase as well. This degrades network capacity, flexibility and scalability. Unlike other networks, MANETs are designed to have minimal manual configuration, low cost of hardware, and tolerance to jamming attacks and mobility [2].

Jamming is caused by high powered devices intentionally designed to attack a wireless network, which is referred to as intentional jamming. Also, jamming can arise unintentionally from non-compatible standards, e.g. 802.11 and 802.15.1 operating in the 2.4GHz ISM band, which is mostly referred to as interference. Another type of jamming in MANETs is caused by denial-of-service (DoS) attacks [3] where malicious stations transmit false messages to consume network resource and starve other stations. To mitigate jamming, first we need to detect it and then avoid using the jammed channel. Under the assumption that such jamming attacks are detectable, stations can dynamically switch from one channel to another [4] as in dynamic spectrum access, where stations search for a new channel when the current operating channel is unavailable.

Due to the reduction of radio cost, it is now easier to fit multiple interfaces in one station, as seen in Wi-Fi devices that work for both 2.4GHz and 5GHz. Therefore, our aim is to exploit this advantage to enhance the network capacity and flexibility of MANETs with the help of Multiple Interfaces of stations operating on Multiple Channels (MIMC) [5] by proposing a new channel assignment scheme. In addition, the channel assignment algorithm should overcome the jamming problem as well.

In this paper, we first formulate the channel assignment problem as graph partitioning problem that minimizes the number of adjacent vertices on the same partition, and then propose a distributed and heuristic channel assignment algorithm called Channel Assignment and JAmmer Mitigation (CA-JAM), because the problem is found to be non-deterministic polynomial-time hard (NP-hard). In CA-JAM, first, each station determines a distinct random channel for all of

☆ Part of this work has been presented in ACM International Conference on Ubiquitous Information Management and Communication (IMCOM) 2014.

* Corresponding author. Tel.: +82 312193618; fax: +82 312191688.
E-mail addresses: zizutg@ajou.ac.kr, zizutg@gmail.com (Y.Z. Jembre), choiyj@ajou.ac.kr (Y.-J. Choi).

[1] Throughout the text, the terms interface and radio are used interchangeably.

http://dx.doi.org/10.1016/j.comcom.2015.09.003
0140-3664/

its interfaces and exchanges beacons on each interface for a simple rendezvous process. To enhance the rendezvous process, algorithms described in [6] and [7] can be used. Up on receiving beacons from neighbors, stations organize the information into two tables: an interface table and a neighbor table. To avoid interference, stations look up the number of neighbors per interface from its interface table and check whether it has multiple links with all neighbors on this interface. If these conditions are satisfied, the station switches to another channel to organize a less congested network and seek more connectivity.

When a channel is jammed, the following steps are performed: (1) neighbors that are exclusive to the jammed interface are selected, i.e., neighbors with a single link; (2) channel information about the unjammed interfaces of these neighbors is inferred; and (3) the channel that is shared among most neighbors is selected and assigned to the jammed interface. Then, the interface switches to the assigned channel to re-establish communication with its neighbors to recover the lost connection due to the jammer presence. CA-JAM is a beacon-based fully distributed channel assignment scheme resistant to jamming attacks, where there is neither control channel nor a central unit. To make our algorithm off-the-shelf 802.11-compatible, we only slightly modified the beacon type that was defined in the IEEE 802.11 standard.

The rest of the paper is organized as follows. In Section 2, related work is presented, and in Section 3, our problem statement is described. In Sections 4 and 5, the proposed CA-JAM algorithm and our simulation results are described, respectively. Our conclusion is presented in Section 6.

2. Related work

The use of multiple interfaces incurs a channel assignment issue. There exist several studies about channel assignment in literature each from different perspective; in this section we assess previous works in this issue. In [8] and [9], the authors provided approaches on how graph theory can be used for channel assignment in MANETs. In [10], graphs are substantially applied to the channel allocation. First, the topology is determined based on the connectivity and interference graph. The interference graph is used to determine link interference, and an optimal algorithm is formulated based on the connectivity graph to reduce the multichannel link interference. Finally, authors proposed an approximation algorithm, because the coloring solution for the formulated graph is NP-hard. Moreover, interference among users is considered when designing channel allocation schemes. In [11], a centralized multi-radio conflict graph is used to model interference among stations, where a channel is assigned using station intelligence to minimize interference throughout the network. In [12], another form of conflict graph called Multi-Dimensional Conflict Graph (MDCG) is proposed to find a possible non-interfering channel assignment.

In [13], the authors proposed a static channel assignment algorithm, where each interface is assigned a distinct channel and that will determine the topology. The assignment strategy is to allocate interfaces in a common neighbor with as many distinct channels as possible such that the interference among connections is minimized. The authors of [16] suggested that one of the multiple interfaces is static while others are dynamic. In this method, HELLO messages are exchanged among the stations over the static interface and the information extracted from this message is used to mitigate interference among stations.

The proposed scheme in [36], strives to minimize interference with the help of channel assignment. This is a greedy assignment which requires all the links in the network as an input. The goal is to maximize the number of links that operate simultaneously. First each link is mapped with the first channel then the upper and lower bound SINR of each link while using that channel is obtained. Following that links are prioritized based on the upper and lower bound SINR on that channel; the link with highest priority will be assigned to that channel and the rest of the links are mapped to the next channel. This process is repeated until all the links are mapped to channel.

Adaptive Dynamic Channel Allocation protocol (ADCA) is a hybrid channel assignment protocol proposed in [25]. Like [16], one interface of each station is static while the others are dynamic. The purpose of the static interface is to enhance throughput between a station and central node whereas the dynamic interfaces are designed to work in on-demand mode. Time is divided into fixed intervals, each having a control and data interval. In the control interval dynamic interfaces negotiate a channel and choose the "least congested channel". Each station has a queue associated with its neighbors and in the data interval the algorithm takes the queue length into consideration to choose to which group of neighbors it should communicate first. In [15], IEEE 802.11 stations obtain neighbor information through scanning and beacon broadcast. Then, they form a local coordination group based on similarity of available channels. The group votes on channels to select one channel as a coordination channel for future channel assignment.

The work in [21] suggests a group-based channel assignment (GCA) algorithm based on a divide and conquer approach. In GCA, stations are classified into a master and slaves, where the master is responsible for gathering new link formation and channel assignment. When a slave node joins the network it will send a request to its neighbors who acknowledge with a reply. However, it is the job of the master station to decide whether the new station's link should be activated or not. If it decides to activate the new link, the master will send a broadcast message to all neighbors about the activation of the new link. In GCA, links are grouped based on the bandwidth requirement, and the groups are organized to form components. Then, links of each component are assigned to different channels. The first and second procedures guarantee a balance whereas the last procedure enhances throughput and fairness.

Another group based channel assignment is found in [33]. In this scheme, stations that are fully connected to one another belongs to one group. Stations that overhear other groups communication are called bridge stations. First each group will be assigned one channel in such a way that there is no collision between neighboring groups. One interface of each station in the group will use that for communication. Bridge stations will form fully connected group on their second interface and that group will be assigned a channel in the same manner. When stations communicate with in the group a Latin square based scheduling is implemented to avoid collision. When stations would like to communicate members of other group than its own then it will forward its data to the bridge station; the group made of bridge stations also employ Latin square method to avoid collision.

The authors in [14] investigate the joint effect of topology control and channel assignment in two stages; first, every station adjusts and checks its link until an undirected graph is constructed; second, every station is assigned a transmission channel. In [17], a joint channel assignment and routing protocol is proposed to minimize the maximum number of l-hop neighbors that share the same channel. Each station builds the network topology from periodical HELLO messages sent through the common control channel (CCC). Stations detect active neighbors using the request-to-send (RTS)/ clear-to-send (CTS) through CCC. Then, they choose a channel that is available for both stations from their list. Other stations update their available channel list upon listening to the RTS/CTS. The use of a CCC and 802.11-like MAC protocol is also found in [18].

Unlike [17], to detect channel availability, stations use spectrum sensing in addition to HELLO messages. One transceiver is used for a CCC whereas others are used for data exchange; Data Transmission reServation (DTS) is used for control packet transmission in order to announce spectrum reservation and transmit power to neighbors.

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003

Table 1
Notations and their description.

C	Number of channels
n_I	Number of interfaces per station
C_{jmd}	Jammed channel number
N_{RC}	New random channel number
u_c^i	Interface i of station u operating on channel c.
$d(u_c^i)$	Degree (i.e. number of neighbors) of station u on interface i and channel c
$S(u_c^i)$	Set of distinct neighbors of station u on interface i and channel c
d_{opt}	Optimal number of neighbors per interface
$k(G)$	Connectivity of graph G
Θ^{CG}	Group of neighbors that share the same channel

Similarly, the work in [24] uses control information to implement channel assignment in MIMC environments. However, they reuse the 802.11 RTS/CTS messages in pursuit of avoiding two-hop interference. The basic steps of this work are summarized as follows: when two stations want to communicate, they exchange RTS/CTS on the control channel and the neighboring stations avoid using the same channel by overhearing the control information.

A channel assignment based on parallel rendezvous scheme with two interfaces was proposed by the authors in [34]. Stations uses hopping sequence to form a link with neighbors. The first interface is used for transmission and uses fast hopping whereas the second interface is used for reception and transmission of control messages such as HELLO packets; it follows slow hopping sequence. Station is required to be synchronized with its 1-hop neighbors with the second interface using the HELLO messages. Channel is divided into time slots, when two stations are in the same slot communication occur.

In [19], the authors proposed an Ad hoc On Demand Distance Vector (AODV) based joint channel assignment and routing algorithm. Similar to [17,18], this method uses a CCC for information control; however, it uses interference indices to weight channels to avoid interference in the network. The authors of [26] categorized channels into control and data classes. This work also modifies the famous AODV protocol to exchange control information. The authors in [23] add an extra routing layer to address channel assignment together with routing. First, each interface of a station is indexed and the channels from the lowest to the highest are assigned to these interfaces. The interface with the lowest index and channel is used for best efforts; i.e. control information includes the channel and interface. This work modifies the traditional Open Link State Routing (OLSR) protocol for the purpose of enhancing the throughput and performing channel assignment. The control information is piggybacked in Topology Control (TC) message of OLSR. The channel assignment is then performed using TC messages which are received from the neighbor based on estimation of available bandwidth of each node.

In [22], channel assignment using particle swarm optimization (PSO) is proposed. This is inspired by a social behavior, e.g. bird flocking or fish schooling. Due to the discrete nature of multi-hop networks, the original PSO could not be directly applied for CA. For this reason, the authors first developed discrete PSO (DPSO) algorithm to form DPSO-CA. The main focus of the work is to minimize interference while maintaining a topology. To preserve a topology, two neighbors must be assigned at least one channel. Interference is avoided by applying crossover and mutation for the initial channel assignment of the network, which is supervised by a central node. A more comprehensive survey on channel assignment can be found in [27].

Another bio-inspired joint work is found in [35]. The authors formulated an optimization problem to get the optimal channel and power allocation of the entire network such that throughput and fairness are enhanced. The problem is complex and NP-hard hard to find the optimal solution, which made the authors to consider the hybridization of two bio-inspired schemes, genetic algorithms (GA) and PSO such that optimal channel and power allocation are achieved. The authors used PSO to find the sub optimal solutions and use the solutions from PSO in GA to find the optimal solution.

Most 802.11-based MAC protocols require the availability of a CCC [17–19,23,24], topological view [13,14,22] or central entity [11,12,21] for channel assignment. Although, CCC, central entity or the whole network could easily be jammed or compromised, none of the above works has taken jamming into consideration. In addition, some of the works solve the channel assignment problem jointly with routing [19,23,26]. However, these works fail to work with multiple routing protocols designed for MANETs. Taking the design principles outlined in [20] into account and also considering the weakness of exiting work we have proposed CA-JAM; the contribution of this work in comparison with existing works is summarized as follows:

- We developed a channel assignment strategy that is scalable and flexible for multi-interface multi-channel MANETs while being totally distributed i.e. no CCC, central entity and no association with network layer for routing. In addition, we develop a scheme that enhance throughput and reduces interference. Our algorithm also improves connectivity by balancing stations that are in the same vicinity and using the same channel. Finally, CA-JAM is different from existing works because it provides an adaptive channel assignment against jamming attacks.[2]

3. Problem definition

For MANETs, the topology of a network can be represented as connectivity graph $G(V, E)$, where a vertex in V represents a station in the network and an edge in E is placed between two vertices (u, v), if they are within the communication range. Let $|V|$ denote the cardinality of V. A conflict graph is consequently defined to account for interference issues. In a conflict graph, vertices represent the communication links and edges are placed between vertices corresponding to interfering links based on the protocol model. However, the conventional conflict and connectivity graph does not capture the MIMC environment. To support our system model, we discuss a multi dimensional conflict graph (MDCG) in the following subsection. The notations used in our system model are described in Table 1.

3.1. Conflict graph

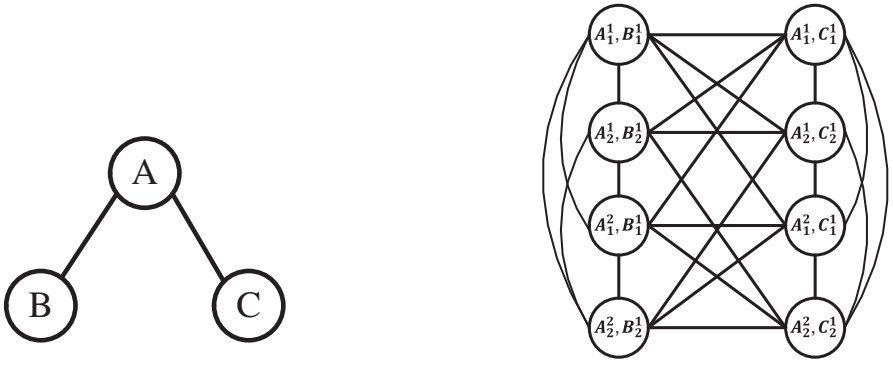
An MDCG for MIMC MANETs is defined in [12]. To describe the conflict relationship among stations, we modify radio-link-channel (RLC) tuples used in [12] to simple-RLC$_M$ (S-RLC_M); where M is the number of tuples created by stations with a common channel and can be calculated as $M \leq \lceil(|V| \times n_I)\backslash 2\rceil$. S-RLC is defined as (u_c^i, v_c^j) and indicates that there is a common channel c between interfaces i and j of stations u and v that forms a link. Using these tuples, we can describe all the possible conflicts in a MDCG. The conflicts can be explained using three events:

- X: stations are interfering in a protocol model.
- Y: stations are associated with the same channel.
- Z: stations that share a common interface at one or two stations.

There is a conflict between tuples when the condition $(X \cap Y \cap \overline{Z}) \cup Z$ is true for two different S-RLCs. The first condition $X \cap Y \cap \overline{Z}$ implies that concurrent transmissions within the same interference

[2] This work is an extension of our previous work [32]. This work provides full description of the working principle of our proposed scheme, the proof on the NP-hardness of the scheme, and further performance evaluation.

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003



(a) Simple graph with 3 stations (b) Resulting MDCG graph

Fig. 1. Conversion from a given graph to conflict graph.

range are not supported, whereas the second condition Z implies that a single interface cannot support multiple transmissions. Let $\overline{S\text{-}RLC} = \{S\text{-}RLC_1, S\text{-}RLC_2, \ldots, S\text{-}RLC_m\}$ $(m \leq K)$ be the set of tuples that transmit concurrently, then an indicator function is defined as follows:

$$f(\overline{S\text{-}RLC}) = \begin{cases} 1 & \text{if there is a conflict in } \overline{S\text{-}RLC}, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

To illustrate how the MDCG works, we assume a network with two available channels and three stations A, B and C. Here A has two interfaces while B and C have only one interface. A is within the transmission range of both B and C; however, B and C are not in the transmission range of each other as shown in Fig. 1a. This network can be mapped into four $S\text{-}RLC$ tuples. For example, tuple (A_1^1, B_1^1) means that station A transmits to B on channel 1 and both stations use interface 1. Therefore, using $(X \cap Y \cap \overline{Z}) \cup Z$ condition and considering all tuples, we obtain the MDCG as shown in Fig. 1b. According to the MDCG in Fig. 1b, there is no conflict in either $\{(A_1^1, B_1^1), (A_2^2, C_2^1)\}$ or $\{(A_2^2, B_2^1), (A_1^1, C_1^1)\}$ in $\overline{S\text{-}RLC}$ set. This implies (1) there will be no interference between links if the network is configured according to one of these two sets and ; (2) there is no interface conflict occurs in the network. Although the MDCG finds the ultimate none interfering tuples, it has several deficiencies as discussed in Section 2. Therefore, we propose an algorithm that can find the non-interfering tuples while avoiding these drawbacks of MDCG. In the following, we formulate the optimization problem.

3.2. Optimization problem

The goal of channel assignment is to optimally allocate the limited number of channels available for MIMC MANETs. If the number of stations is less than the available channels or if the number of interfaces per station is equivalent to the available channels, then the optimal channel assignment can be achieved easily by exhaustive resource mapping. However, in most practical scenarios, the available channels are less than the number of stations and more than the number of interfaces. Minimizing the number of stations that are using the same channel minimizes interference and enhances the throughput; however, it affects connectivity $k(G)$. Therefore, we need to identify the optimal number of stations that are with in close proximity and share the same channel while there is at least one path for any source destination pair in the network. Also no two interfaces of one station can have the same channel and there is no conflict according to the MDCG, i.e., two links on the same interference range do not transmit simultaneously. Mathematically, the optimization problem can be stated as follows:

$$\begin{aligned} & d_{opt} = \{\min \ \ d(u_c^i)\} \\ & \text{subject to} \\ & u_c^i \neq u_c^j, \qquad \forall (i,j) \in N^I \\ & k(G) \geq 1, \\ & f(\overline{S\text{-}RLC}) \leq 0, \\ & k(G), f(\overline{S\text{-}RLC}) \in \{0,1\} \\ & i, j, c \in \{0,1,2,\ldots\} \end{aligned} \quad (2)$$

where

$$k(G) = \begin{cases} 1 & \text{if there is at least one path for any source} \\ & \quad \text{and destination,} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Since the Finding the minimum degree of a vertex on each of its interfaces is equivalent to finding the optimal number of neighbors (d_{opt}) that one station should have on each of its interfaces. The first constraint defines that two interfaces of one station cannot have the same channel. The second constraint means that the connectivity of the graph should always be greater than or equal to one. This enforces that there should always be one path for any (source, destination) pair. The third constraint states that the final solution to the optimization problem should not contain conflicting tuples; i.e., interference should not be caused while trying to solve the optimization problem. Therefore, if we can find the optimal number of neighbors that one station has to have on each of its interfaces, then by limiting each interface to that number, we can obtain the optimal throughput and connectivity because that is the optimal number of stations that should share the same channel. Like the NP-hardness of channel assignment problems in many literatures [10,28,29], the channel assignment of our work is also NP-hard for the MIMC ad-hoc networks. Since the domain of all the variables of this optimization problem is in integer, the nature of this optimization problem is of integer linear programming, which is NP-hard. The proof of the NP-hardness of the optimization problem is given as follows.

Theorem 1. *It is NP-hard to find the optimal number of neighbors at each interface.*

Proof. We want to divide the network such that every group has the same number of stations and there is connectivity between any two stations $(k(G) \geq 1)$. Let $d(u_c^i)$ be the degree of a vertex of station u on interface i and channel c, and let d_{opt} be the optimal degree that every station has, i.e., the minimum number of neighbors of one station that share the same channel on a single interface. We show that it

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003

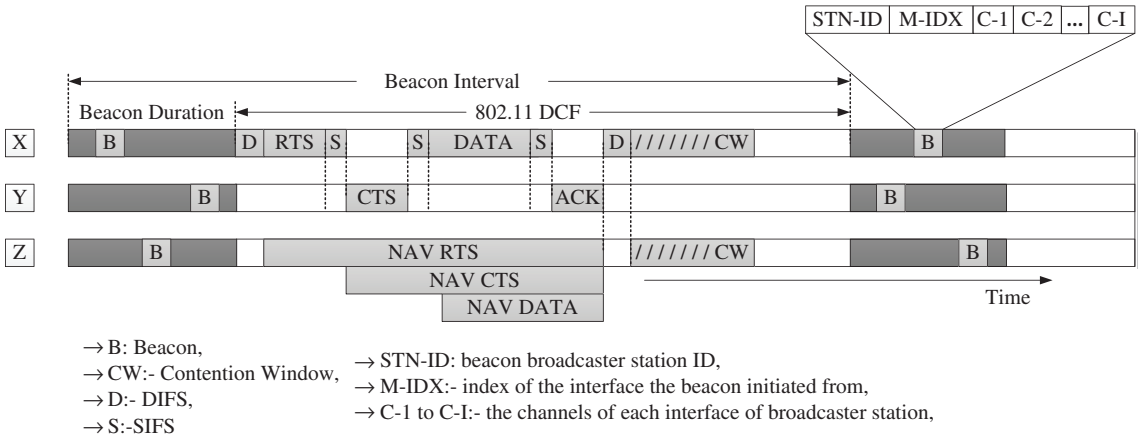


Fig. 2. Beacon and data transmission in CA-JAM.

is NP-complete to find d_{opt} using the reduction from graph partitioning problem [30,31,31]. The graph partitioning problem states that given a graph $G(V, E)$ and a constant k, it is NP-complete to induce G into k sub graphs $G_1, G_2, \ldots, G_k$ where the vertices of the induced sub graphs are equal, while minimizing the disconnection, i.e., capacity of edge cut, between sub graphs. Suppose we find d_{opt}, which implies $d(u_c^i)=d_{opt}$ for all interfaces of one station on any channel. This means that the number of stations on a certain channel is equal to the degree of one station plus the station itself $(d_{opt}+1)$. This solution gives a valid graph partitioning with minimum edge cut. Therefore, it is NP-complete to find an optimal degree for any d_{opt} and $d(u_c^i)$. □

3.3. Design considerations

Before explaining our CA-JAM algorithm, we discuss some features that should be considered to design a good channel assignment scheme. Later, the effectiveness of the proposed CA-JAM will be evaluated based on these principles.

- Interface-channel mapping: in the MIMC MANETs, mapping an interface to a channel is not a simple task. It could create a disconnected network if interfaces of communicating neighbors have no common channel, i.e., neighbors would not be able to rendezvous.
- Connectivity: channel assignment in MANETs is about having a connected network. Because there is no central unit to oversee the delivery of a packet from a source to a destination, the only way to achieve successful data delivery is to guarantee a path from any source to any destination.
- Interference and jamming avoidance: interference could be caused intentionally by stations that want to disrupt an ongoing communication (jamming) or unintentionally by stations that are not aware of an ongoing communication (interference). A good channel assignment scheme should be able to avoid both types of interference.

4. CA-JAM algorithm

In our system, stations exchange interface-to-channel mapping information using beacons that are broadcasted by every station periodically. The data communication between consecutive beacons follows the IEEE 802.11 protocol. Fig. 2 depicts beacon and data transmission in our system. It only shows the operation of one interface of three stations operating on the same channel. Time is divided into Beacon Interval (BI), which consists of Beacon Duration (BD) and 802.11 DCF duration. In BD, stations randomly choose the time to broadcast a beacon. We only modify the beacon message to include the interface/MAC index sending the beacon and the channels that are assigned to all the interfaces of a station, which makes CA-JAM easily implementable using off-the-shelf 802.11 interfaces. The received beacon is organized in two look-up tables, the *Neighbor_Table* and *Interface_Table*. When extracting the information from the beacon, if *M-IDX* is 2 the channel assigned to the sending interface is *C-2*. The rest *C-i's* are used to re-establish a connection with the sending station in case this connection is lost. The detailed discussion on how to use the *C-i's* to re-establish the lost connection is discussed in the following subsections.

First, the *Neighbor_Table* is a collection of tuples with the following information.

- `Neighbor_ID` stores the information collected from STN-ID field of the beacon.
- `MAC_Index` stores information extracted from M-IDX of the beacon. Because a neighbor has multiple interfaces, `MAC_Index` identifies the MAC of the interface that sent the beacon.
- `NbrChannel_i`$^{(i\epsilon\{1,\ldots,I\})}$ stores the neighbor's channel number for each interface and is collected from C_i fields of the beacon, where $(i\epsilon\{1,\ldots,I\})$.
- `Expiry_Time` stores the maximum time a tuple remains relevant after it is created.

Second, the *Interface_Table* is made up of the following tuples:

- `Nbrs_on_Interface_i`$^{(i\epsilon\{1,\ldots,I\})}$ stores the list of neighbor IDs that are exclusive to interface i for I interfaces. This helps a station track the number of neighbors on each interface and their identities.

These tables are updated on three occasions: (1) when the channel of an interface switches; (2) when the information has expired and; (3) when a beacon is received. Note that a neighbor might be found on more than one interface, which means there are multiple links between any two stations. CA-JAM is solely based on the above look-up tables and composed of three modules: rendezvous, interference avoidance, and jammer mitigation, which will be explained in the following subsections.

4.1. Rendezvous

This module covers the channel assignment during the MIMC MANET initialization or when a station joins the network for the first time. It is assumed that interfaces are indexed from 1 to n_I. Therefore, the station iterates through all of its interfaces and assigns each one a random channel that is different from others. This will enable stations to satisfy the first condition of the optimization problem, i.e., two interfaces of one station cannot have the same channel $(u_c^i \neq u_c^j)$.

Table 2
Rendezvous algorithm.

Input- C: number of channels, n_I: number of interfaces

```
Procedure Rendezvous()
| 1: for i = ( 1 to n_I ) do
| 2:   channel_of_u^i = newRandChNum()
| 3:   i = i + 1
| 4: end for
| 5: while( beacon_duration )
| 6:  for i = ( 1 to n_I ) do
| 7:   while( d(u_c^i) == 0 )
| 8:    channel_of_u^i = newRandChNum()
| 9:    i = i + 1
| 10:   end while
| 11:  end for
| 12:  beacon_duration = next_duration
| 13: end while
End procedure
```

Table 3
Random channel generator algorithm.

Input- C_{jmd}: jammed channel number $N_{RC} = 0$: new random channel number

```
Procedure newRandChNum()
| 1: N_RC = random( 1,...,C )
| 2: for i = ( 1 to n_I ) do
| 3:  if ( N_RC == channel_of_u^i OR N_RC == C_jmd ) then
| 4:   N_RC = random( 1,...,C )
| 5:   i = 1
| 6:  end if
| 7:  i = i + 1
| 8: end for
| 9: return N_RC
End procedure
```

The rendezvous and random channel generator algorithms are given in Tables 2 and 3, respectively.

After all interfaces have a distinct channel according to lines 1 to 3 of the rendezvous algorithm, the station broadcasts beacons on all interfaces and listens to its neighbors' beacons. If a station hears at least one beacon from neighbors, then a rendezvous is completed and the station starts to update the tables. However, if there is no beacon for the whole beacon duration, which means there is no neighbor on that interface, the station waits until the next beacon duration, assigns a new channel, and broadcasts its new assignment while listening again (lines 5 to 13 of the rendezvous module). In other words, the station will stop rendezvous process when it receives a beacon from other station, which mean there is at least one node in that channel. Line 3 in the random channel module ensures that the new channel is not jammed or has already been assigned to another interface. This process is repeated until there is at least one neighbor on each interface, i.e., $d(u_c^i) \geq 1$. Through this process, the second condition of the optimization problem is fulfilled, i.e., $k(G) \geq 1$. Unless the distance between one or a group of stations is much longer than the remaining network, there will be no disconnection. This is done by counting the entities of `Nbrs_on_Interface_i` from the *Interface_Table*. In other words, if the count of `Nbrs_on_Interface_i` is zero, it means there is no neighbor on that interface.

4.2. Interference avoidance

Now that the network has been established through rendezvous, all stations in the network have at least one or more neighbors on each of their interfaces. Let Θ^{CG} be the group of stations that reside on the transmission range of each other and use the same channel to form a communicating group. However, for successful communication , the Θ^{CG} group has to overcome interference that is caused by one of the following scenarios:

Table 4
Interference avoidance algorithm.

Input- $d(u_c^i)$: degree of a station u on interface i, N^I: number of interfaces, $S(u_c^i)$: set of distinct neighbors on interface i,

```
Procedure interferenceAvoidance()
| 1: for i = ( 1 to N^I ) do
| 2:   S(u_c^i) = getDistinctNbrs(i)
| 3:   nbrsOn_Int_i = Interface_Table.getNext()
| 4:   d(u_c^i) = nbrsOn_Int_i.sizeOf()
| 5:   if ( d(u_c^i) > d_opt AND S(u_c^i) == ∅ ) then
| 6:    channel_of_u^i = newRandChNum()
| 7:   end if
| 8:   i = i + 1
| 9: end for
End Procedure
```

- First, the hidden and exposed terminal problems occur because they are common among the IEEE 802.11 networks.
- Second, interference or congestion occurs, if the number of stations in Θ^{CG} is high.
- Third, interference occurs among Θ^{CG} if there is any station that has a neighbor within the interference range that uses the same channel ($f(\overline{S\text{-}RLC}) = 1$).

Because stations that belong to the same Θ^{CG} listen to the probe messages broadcasted in the group, the first type of interference is easily avoided using the RTS/CTS mechanism. The second type of interference is caused by capacity overload due to the existence of many stations in the same Θ^{CG} or channel. Therefore, to avoid such interference, a station needs to check if the number of its neighbors on each specific interface $d(u_c^i)$ is greater than d_{opt}. Here, d_{opt} is the maximum number of neighbors a station can have on each of its interfaces optimally. For that matter, we compare if the current number of neighbors on one interface is greater than d_{opt}. For interfaces with $d(u_c^i) \geq d_{opt}$, stations check whether there are multiple links with all neighbors on this interface (i.e. if $S(u_c^i) = \emptyset$). When the condition holds, $S(u_c^i) = \emptyset$, the station re-assigns a new distinct channel to that interface such that the current channel is less congested and new connections are created. This enhances the connectivity. Table 4 shows the pseudo code of the interference avoidance module, where lines 2 to 4 get the size of neighbors and the set of distinct neighbors of an interface, and lines 5 to 7 check the conditions (that the number of neighbors on this interface has exceeded d_{opt} and all neighbors in this interface have multiple links with this station) and when the condition is true station assign a new channel to this interface. Otherwise, the station will remain on the same channel because connectivity is more critical in MANETs. $S(u_c^i)$ is obtained by cross-referencing both tables. Station matches the `Neighbor_ID` with the `Nbrs_on_Interface_i` for $i \epsilon \{1, \ldots, I\}$, if `Neighbor_ID` exists in more than one interface , it means neighbor has multiple link with this station so it will not be added to $S(u_c^i)$. This implies $S(u_c^i) = \emptyset$, means all neighbors on interface i has multiple links with this station. The module to select distinct channels is shown in Table 5, where lines 3 to 9 count the number of interfaces on which neighbors exist and lines 10 to 12 check if the count is one i.e, if a neighbor is exclusive to the interface.

Eventually, when this process is repeated by every station, the number of neighbors on every interface converges to the minimum number that could communicate to that interface $d(u_c^i) = d_{opt}$. Thus the goal of the optimization problem is accomplished as stated in Section 3. This process implicitly avoids the primary interference; this will be discussed using an example in Section 4.4.

4.3. Jammer mitigation

CA-JAM is able to switch channels to re-establish a link against jamming. When an operating channel of an interface is jammed, the

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003

Table 5
Distinct neighbor selection algorithm.

Input- i_{num}: Interface Number

```
Procedure getDistinctNbrs(i_num)
| 1: count = 0
| 2: while ( Neighbor_Table.hasNext() )
| 3:   nbr <- Neighbor_Table.getNext()
| 4:   for i = ( 1 to N^I ) do
| 5:     nbrsOn_Int_i = Interface_Table.getNext()
| 6:     if ( nbr.isIn_List(nbrsOn_Int_i) )
| 7:       Increment count
| 8:     end if
| 9:   end for
| 10:  if ( count == 1 )
| 11:    S(u_c^{i_num}) <- nbr
| 12:  end if
| 13:  count = 0
| 14: end while
| 15: return S(u_c^{i_num})
End Procedure
```

Table 6
Jammer mitigation algorithm.

Input- C_{jmd}: jammed channel number, /∗ flag is used to indicate whether BCH and CCH can be used or not. If flag = 0 at least one of the can be used if flag = 1 then the jammed channel should be assigned new random channel∗/ flag = 0 first it is set to 0

```
Procedure jammingMitigation()
| 1: for i = ( 1 to n_I ) do
| 2:  if ( C_jmd == channel_of_u^i ) then
| 3:    getBackupCandidateCH(i)
| 4:    channel_of_u^i = BC(u^i)
| 5:    for j = ( 1 to N^I ) do
| 6:      if ( channel_of_u^i == channel_of_u^j and flag == 0 ) then
| 7:        new_CH = CC(u^i)
| 8:        j = 1
| 9:        flag = 1
| 10:     else if ( flag == 1 ) then
| 11:       channel_of_u^i = newRandChNum()
| 12:     end if
| 13:   end for
| 14:  end if
| 15: end for
End Procedure
```

station selects a set of neighbors that are distinct to the jammed interface $S(u_c^i)$ from the table, i.e., stations will try to reconnect with neighbors that share a single link rather than multiple ones. This is described in Table 6. In line 2, it finds which channel is jammed and in line 3, the preparation of backup and candidate channels for that interface is requested by invoking the module as described in Table 7. Similar to the interference avoidance module, members of $S(u_c^i)$ are selected by matching the `Neighbor_ID` with the `NbrID_on_Interface_i` for $i\epsilon\{1,\ldots,I\}$. Then, for each neighbor in $S(u_c^i)$, stations compare the jammed channel C_{jmd} to the neighbors' channels. When a match is found , i.e., $C_{jmd} = c_i$, $i\epsilon\{1,\ldots,I\}$ of the neighbor, channels c_{i+1} and c_{i+2} will be added to the list of eligible backup (L-BCH) and candidate (L-CCH) channels, respectively. When $C_{jmd} = c_{N_{I-1}}$, then c_{n_I} and c_1 will be used in the respective lists. If $C_{jmd} = c_{n_I}$, then c_1 and c_2 will be used. This is done through the backup and candidate channel module, lines 2 to 14. According to lines 15 and 16, the most common channel from $S(u_c^i)$ become backup (BCH) and candidate (CCH) channel using L-BCH and L-CCH. [3] When the operating channel of an interface of a station is jammed due to the presence of neighbor users, it independently switches to the BCH. If the BCH is not available, that interface switches to the CCH. In the circumstance when both BCH and CCH channels are jammed, a new channel other than the BCH and CCH will be assigned to the interface using the rendezvous process.

Table 7
Backup and candidate channel selection algorithm.

Input- i_{num}: Interface Number $BCH(u^i) = 0$: backup channel interface i, $CCH(u^i) = 0$: candidate channel interface i

```
Procedure getBackupCandidateCH(i_num)
| 1: S(u_c^{i_num}) = getDistinctNbrs(i_num)
| 2: while ( S(u_c^{i_num}).hasNext() )
| 3:   distNbrCH <- S(u_c^{i_num}).getChannel()
| 4:   if ( i_num != n_I and i_num != N_{I-1} )
| 5:     L-BCH <- distNbrCH_{i_num+1}
| 6:     L-CCH <- distNbrCH_{i_num+2}
| 7:   else if ( i_num == N_{I-1} )
| 8:     L-BCH <- distNbrCH_{n_I}
| 9:     L-CCH <- distNbrCH_1
| 10:  else if ( i_num == N_I )
| 11:    L-BCH <- distNbrCH_1
| 12:    L-CCH <- distNbrCH_2
| 13:  end if
| 14: end while
| 15: BCH(u^{i_num}) = L-BCH.getFrequentCH()
| 16: CCH(u^{i_num}) = L-BCH.getFrequentCH()
end Procedure
```

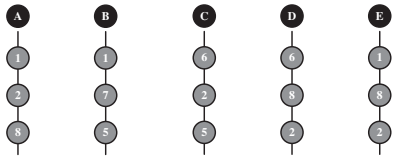
For instance, for stations with three interfaces, if the jammed channel of a neighbor is the first interface, then the channels that its neighbor uses for its second and third interfaces will be added to the list of eligible backup and candidate channels, respectively. Then from the list of eligible backup and candidate channels, the station selects the most dominant ones from both lists and makes them the BCH and CCH for the jammed interface. Next, the station updates the *Neighbor_Table* and assigns the BCH to its interface. If this is also jammed, it tries the CCH. In case both are not available then a random channel will be assigned according to the rendezvous process.

4.4. Example of CA-JAM algorithm

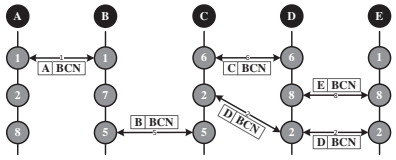
The CA-JAM algorithm is described using an example. We first describe rendezvous, and then explain interference avoidance and jammer mitigation, concurrently. Assume a network of five stations, each with three interfaces. We chose the number of interfaces to be three, because it is suitable to show the working principle of CA-JAM with backup and candidate channel, as discussed below. The simulation follows this as well. Only one neighbor per interface is allowed ($d_{opt} = 1$). First, each station selects a distinct random channel for all its interfaces as depicted in Fig. 3a. When stations pick the same channel with their interfaces and exchange beacons for rendezvous, a communicating group Θ^{CG} is formed. Fig. 3b shows the processes of beacon exchange and rendezvous. For instance, station A on channel 1 with its interface 1, A_1^1, and station B on channel 1 with its interface 1, B_1^1, form communicating group Θ^{CG}. Similarly, other stations form more groups: $\Theta^{CG} = \{(A_1^1, B_1^1), (B_5^3, C_5^3), (C_6^1, D_6^1), (D_8^2, E_8^2), (C_2^2, D_2^3, E_2^3)\}$. During beacon reception, each station updates its *Neighbor_Table* a nd *Interface_Table*. In this paper, we only show the tables of stations B and D due to the lack of space. Fig. 3c demonstrates the update procedure applied for stations B and D.

As a result of assignment and rendezvous process, station D has more neighbors on its interface 3 than it is allowed. Station D is aware of the situation using the information from the *Interface_Table*. In the mean time, suppose channel 1 between station A and B is jammed as depicted in Fig. 3d. The next step for station D is to check whether there are multiple links with all neighbors on interface 3. This condition exists because, station D is connected with C and E through interface 1 and 2, respectively. Therefore, station D switches

[3] Both BCH and CCH might be considered as backup channels. In our system where we have three interfaces, the BCH and CCH of a certain interface are the two channels used in the other two interfaces.



(a) Stations select distinct random channels.



(b) Stations exchange beacons for rendezvous.

Neighbor Table of B

Nbr ID	MAC Index	NbrChannel 1	2	3	Expire Time
A	1	1	2	8	0.5
C	3	6	2	5	0.5

Interface Table of B

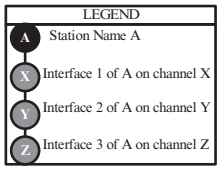
NbrID_on_Interface 1	2	3
A	0	C

Neighbor Table of D

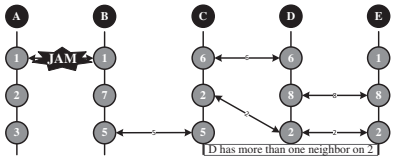
Nbr ID	MAC Index	NbrChannel 1	2	3	Expire Time
C	1	6	2	5	0.5
E	2	1	8	2	0.5

Interface Table of D

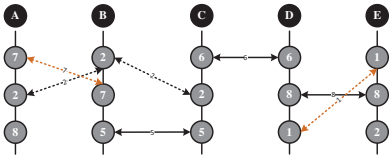
NbrID_on_Interface 1	2	3
C	E	C
0	0	E



(c) *Neighbor_Table* and *Interface_Table* update from stations B and D. D has more than one neighbor on 3^{rd} interface.



(d) Channel 1 between stations A and B is jammed and station D has more than allowed neighbors on interface 2.



(e) Stations A and B avoid jamming attack; Since station D has multiple links with neighbors C and E, D avoids congestion.

Neighbor Table of B

Nbr _ID	MAC _Index	NbrChannel 1	2	3	Expire _Time
A	2	7	2	8	0.5
C	3	6	2	5	0.5

Interface Table of B

NbrID_on_Interface 1	2	3
A	A	C

Neighbor Table of D

Nbr _ID	MAC _Index	NbrChannel 1	2	3	Expire _Time
C	1	6	2	5	0.5
E	2	1	8	2	0.5

Interface Table of D

NbrID_on_Interface 1	2	3
C	E	0

(f) *Neighbor_Table* and *Interface_Table* update from stations B and D.

Fig. 3. The procedure followed in the CA-JAM algorithm with three interfaces per station.

to another channel to *avoid interference* and most likely switches to a channel that is common with one of its neighbors. Meanwhile, station B finds which interface of station A lost a link. Using this information, station B adds the channel of the second and third interfaces to the backup and candidate channel list (L-BCH = { 2 } and L-CCH = { 8 }). Because there is no other neighbor on this channel at the time of jamming, both lists only contain a single element, which implies these channels directly become backup $BCH(B^1) = 2$ and candidate $CCH(B^1) = 8$ channels. Hence, station B switches its interface one to channel 2. It is important to note that, if station D had not left channel 2, the interference in that channel could have been severe. Although it is not shown here station A follows the same procedure as B; therefore, interface 1 of station A switches to channel 7 to mitigate the jammer effect. As depicted in Fig. 3e, CA-JAM results in a more connected and less interfered network at the end of channel assignment and jammer mitigation. Finally, Fig. 3f shows the updates of the *Neighbor_Table* and *Interface_Table* for stations B and D.

When the above process is performed repeatedly, neighbors per interface will be equal to one. This shows that CA-JAM can find non-interfering tuples, similar to the MDCG, without the help of a central entity or use of CCC and manages to mitigate jamming. The CA-JAM algorithm fulfills the design considerations: interface-channel mapping, connectivity, and interference avoidance. However, not every station transmits and receives all the times. Thus, it is not important to have only one neighbor per each interface just to find non-interfering tuples. In addition, stations can benefit from having a few neighbors rather than only one. Therefore, it is essential to determine how many neighbors per each interface (d_{opt}) is sufficient. In the next section, we will determine and show that one neighbor per interface is not the ideal solution through simulation.

5. Performance evaluation

In this section, we discuss experimental results obtained from the OPNET simulator. Table 8 presents our simulation parameters. As per traffic, each station is configured with three traffic types and any station can choose to transmit at any time unless its beacon duration time. During transmission time stations transmit one packet of each traffic type; however, the next transmission time is determined by traffic distribution. Unless specified otherwise, the distribution of packet inter arrival time remains uniform for the entire evaluation. In constant distribution the inter arrival is fixed whereas in uniform distribution the inter-arrival time is varied between the min and max value.Since we modified 802.11a standard in OPNET, to show the improvement by using multiple interfaces, we compared the single-interface scenario with the multi-interface using the legacy IEEE 802.11a protocol. In Fig. 4, we present the network throughput as a function of the number of stations for the IEEE 802.11a protocol

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003

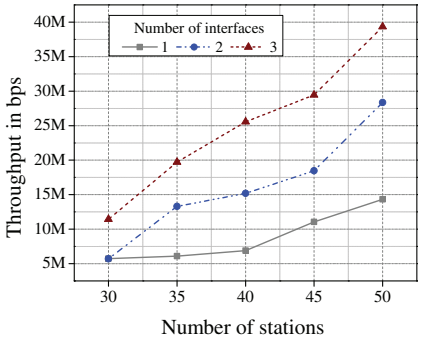


Fig. 4. Throughput comparison of a single and multiple interfaces of IEEE 802.11 stations as the number of stations increases.

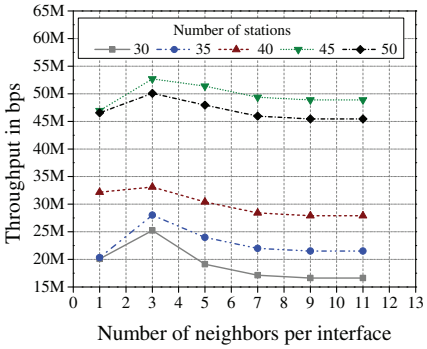


Fig. 5. Throughput as the number of neighbors per interface increases for various numbers of stations.

Table 8
Simulation parameters.

Parameters	Values
Station type	802.11a (can be 802.11 n or ac)
Network deployment area	800 m x 800 m
Transmission range	150 m
Mobility	Random way point
Number of channels	6
Number of interfaces	3
Simulation time	5 min
Routing protocol	AODV
Traffic Type	Email, VoIP, Low resolution video
Traffic generation pattern	Uniform in [1,110]
Data rate	54 Mbps
Ground speed	5 m/sec
Beacon interval	0.5 s
Beacon duration	0.05 s
Jammer transmission power	20 W
Jammer packet inter-arrival time	1 ms
Jammer packet size	1024 bits

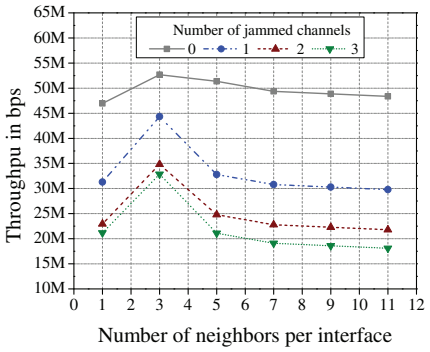


Fig. 6. Throughput as the number of neighbors per interface increases for various numbers of jammed channels (45 stations).

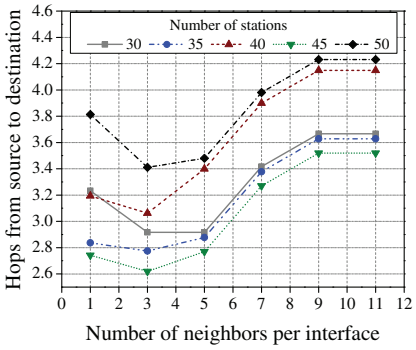


Fig. 7. Number of hops a packet travels from a source to a destination as the number of neighbors per interface increases.

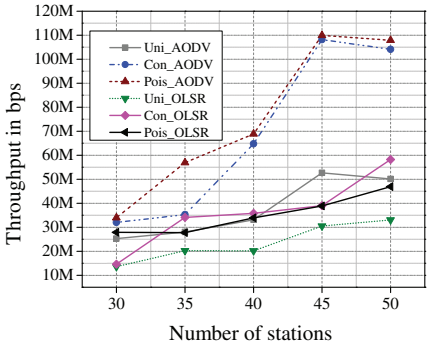


Fig. 8. Throughput of CA-JAM for uniform, constant, and poisson distributions with OLSR and AODV routing protocols..

to show the impact of using multiple interfaces. When stations were equipped with multiple interfaces, the network throughput increased compared with a single interface. This ensures the use of multiple interfaces increases the capacity of the network.

Fig. 5 depicts network throughput as a function of the number of neighbors per interface. The result showed that the throughput was maximum when the number of neighbors was three. It is interesting to note that according to this result, in MANETs, increasing the number of station for a given area did not always provide better throughput, i.e., the throughput of 45 stations was better than that of 50 stations as shown in Fig. 5.

To see the effect of jamming attack on the ideal neighbor number per interface of the network, we kept the number of stations 45 and varied the number of jammed channels. The results are compared in Fig. 6. As the number of jammers increased from zero to three, the ideal number of neighbors per interface remained three; however, the throughput decreased as the number of jammers increased. This explains that CA-JAM adapted to jamming attacks very well. To validate that a MIMC MANET benefits from assigning the ideal number of neighbors per interface, we measured the average number of hops taken by data packet before it reach its destination (see Fig. 7). Moreover, the connectivity of the network was better when the number of neighbors per interface was three ($d_{opt} = 3$). From Figs. 5–7, we have learned that the network performs better when the ideal number of neighbors per interface is three. Therefore, for the remaining evaluation, the number of neighbors per interface for CA-JAM was set to three, $d_{opt} = 3$.

Then we evaluated the performance of CA-JAM for different traffic generation patterns and routing protocols as shown in Fig. 8. Intuitively, we had chosen AODV as a routing protocol for CA-JAM simulation, since it incurs less routing information exchange. We considered the traffic patterns as Poisson and Constant distribution with mean

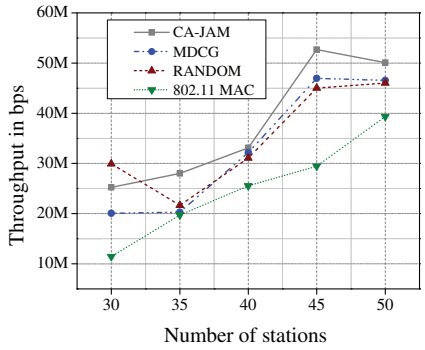


Fig. 9. Throughput comparison of CA-JAM with other schemes, as a function of number of stations.

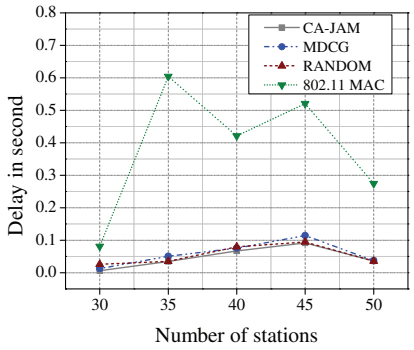


Fig. 11. Delay comparison of CA-JAM with other schemes, as a function of number of stations.

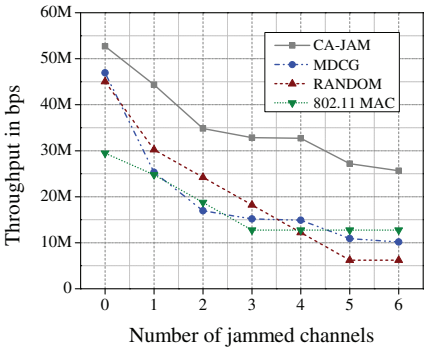


Fig. 10. Throughput comparison of CA-JAM with other schemes, as a function of number of jammed channels (45 stations).

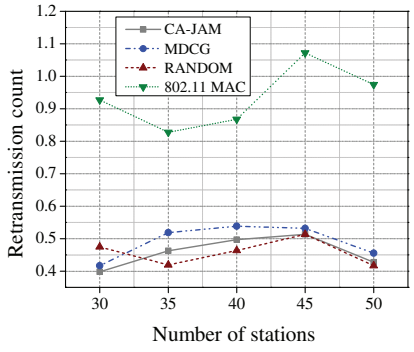


Fig. 12. Retransmission caused by CA-JAM compared with other schemes, as a function of number of stations.

2 as well as Uniform distribution during interval [1,110]. We used routing protocols, AODV and OLSR. When we compared the routing protocols under the same traffic pattern, AODV showed a better performance than OLSR in all traffic distributions. This motivated us to choose AODV as a routing protocol for most of the simulation. However, CA-JAM has flexibility to work with any routing protocol . On the other hand, we simulated with different traffic types to see if CA-JAM reacts to load variation. Fig. 8 evidently shows that CA-JAM can adapt to various traffic types, which shows how CA-JAM could scale as the load in the network increases or decreases. Thus, CA-JAM is proved to work under various traffic loads and routing protocols, which shows its flexibility and scalability.

The performance of CA-JAM was compared with other algorithms for multi-interfaced stations. We compared CA-JAM to MDCG as well as the IEEE 802.11a standard. This is because MDCG captures the MIMC network environment compared to the rest of works mentioned in Section 2. Besides, CA-JAM was also compared with random assignment to show that simple channel assignment does not give the best solution. For the random channel assignment, we modified the IEEE 802.11 system to assign channels randomly. We compared their throughput, number of re-transmissions, and delay. As shown in Fig. 9, the throughput of CA-JAM was higher than the other schemes. While random assignment showed a better result for a small number of stations, it was neither consistent nor better than CA-JAM as the number of stations increased. In addition, this does not mean random is better than CA-JAM in terms of over all performance, which will be shown when we discuss jamming, delay and retransmission afterwards.

Following that, we measured the performance of all channel assignment schemes under jamming attack; jammers jam the network every 1ms and send a packet of 1024 bits. As shown in Fig. 10, the throughput of CA-JAM was always highest and decreased gradually as the number of jammed channels increased. Since jammers can not jam the entire area and due to the data transmitted between consecutive jamming trials, some throughput was still achievable even though all six available channels were jammed. In fact, the throughput of CA-JAM was twice higher than the other schemes, when the network was under jamming attack.

We also compared the delay performance of CA-JAM with other schemes. As depicted in Fig. 11, the delay of CA-JAM was always stable and low, whereas the delay of IEEE 802.11 was very high and unstable because there is no means of controlling interference in 802.11 system. When the number of stations are varied, 802.11 system only follows the network setup, when the setup is suitable the network is more connected and less interfered, which leads to lower delay. On the other hand, in unsuitable network set up stations in 802.11 interfere most of the time since they reside in the same channel which leads to transmission delay due to an unsent packet. This is not the case for MDCG and CA-JAM because these schemes have interference avoidance mechanism where as in random scheme stations reside in different channel, which makes it less prone to delay compared to 802.11 systems. Moreover, the delay of CA-JAM only increased by a very small margin, even when the number of stations increased. In a disconnected network, the number of re-transmission attempts becomes very high, because stations did not have many paths. In Fig. 12, the number of re-transmission attempts showed similar behavior with the delay. These results confirmed that the network achieved better connectivity and exhibited smaller interference under the CA-JAM assignment. In Fig. 9 the random assignment had a favorable channel selection when the number of nodes as small. However, the throughput was enhanced at the cost of higher delay and retransmission as shown in Figs. 11 and 12 respectively. In addition, random assignment exhibits a lower performance under jamming attack. This means that, even when random selection has the favorable conditions, the overall performance of CA-JAM is better in terms of

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003

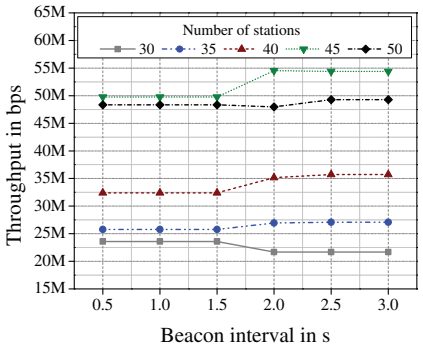


Fig. 13. Throughput as a function of beacon interval.

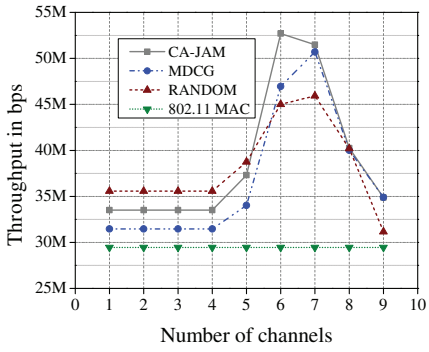


Fig. 14. Throughput as a function of number of available channels.

delay, retransmission and throughput as well as under jamming attack.

Finally, since our work was based on beacon exchange and multiple channels, we investigated the effect of these parameters on the network throughput. Although a longer beacon interval provided some benefit to the network with a higher number of stations, it did the opposite for a network with a smaller number of stations, as shown in Fig. 13. This means, the overhead created by short beacon interval is small. The longer intervals results in inconsistency rather than smaller overheads. Therefore, the selection of making the beacon interval 1 ms was a desirable choice.

In addition, multiple channels with the right channel assignment improved the capacity of MANETs. However, even with a good channel assignment scheme, too many channels did not always create a better throughput. This was because stations took more time for rendezvous. Even when a rendezvous was achieved it was done with small number of neighbors per interface because interfaces of stations were distributed over many available channels. Fig. 14 shows the effect of number of channels in the network. For CA-JAM, the throughput increased until the number of channels was six, and then it started to decrease again. For the MDCG and random schemes, the highest throughput was achieved when the number of channels was seven. Despite the fact that MDCG and random assignments had a maximum throughput at this number of channels, the throughput of both schemes is still lower than the proposed CA-JAM. However, when the number of channels is fewer than the number of interfaces, the random scheme outperforms CA-JAM. Although the available channels cannot be able to accommodate any more partitioning, due to the working principles of CA-JAM, it requires stations to change channels frequently such that there is a balanced assignment, which is not the case in the random scheme. Therefore, CA-JAM is suitable for MANETs where the number of channels is always greater than the number of interfaces.

6. Conclusion

MANETs can be easily deployed to provide essential help in a military and disaster relief networks. However, the presence of jammers, harsh environment of war zone or the absence of central entity, such as access point, affect the performance of such networks. The performance of MANETs can be enhanced by deploying stations with multiple interfaces that operate on multiple channels. While the network benefits from multiple interfaces, it still had limitations due to poor channel utilization. To overcome this issue, we proposed CA-JAM algorithm, which is a channel assignment scheme that considers jammer mitigation as well. Our simulation results showed that CA-JAM provided higher throughput with reasonable delay and re-transmission. The results confirmed that the proposed algorithm could greatly enhance the network performance; thus making MANETs feasible to use in any desired environment.

Although our focus has been channel assignment and current routing protocols are designed for single interfaced MANETS, there is a room to enhance the performance by jointly integrating it with a routing protocol. Our future work will integrate channel assignment with routing to achieve a better performance and fully utilize MIMC feature. Also, we plan to implement our work with off the shelf 802.11 products, to test the performance of CA-JAM in the real environments.

Acknowledgment

This research was supported by the Ministry of Science, ICT and Future Planning (MSIP), Korea, under the Global IT Talent support program (IITP-2015-R0618151003000102) supervised by the Institute for Information and Communication Technology Promotion (IITP).

References

[1] Part 11: wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: high-speed physical layer in the 5 ghz band, 2000, ISO/IEC 8802-11:1999/Amd 1:2000(E); IEEE Std 802.11a-1999, pp. i–83.
[2] J. Redi, R. Ramanathan, The DARPA WNaN network architecture, in: Military Communications Conference, 2011 - MILCOM 2011, 2011, pp. 2258–2263.
[3] G. Noubir, On connectivity in ad hoc networks under jamming using directional antennas and mobility, Wired/Wireless Internet Communications, Springer Berlin Heidelberg, 2004, pp. 186–200.
[4] I.F. Akyildiz, W.-Y. Lee, M.C. Vuran, S. Mohanty, Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey, Comput. Netw. vol. 50 (2006) 2127–2159.
[5] N.V.M. Simone, M. Zorzi, Resource allocation in multi-radio multi-channel wireless ad hoc networks, Online avaialble: http://www.academia.edu/2934172/Resource_Allocation_in_Multi-Radio_Multi-Channel_Wireless_Ad_Hoc_Networks.
[6] R. Paul, Y.Z. Jembre, Y.J. Choi, Multi-interface rendezvous in self-organizing cognitive radio networks. in dynamic spectrum access networks, in: IEEE International Symposium on (DYSPAN), 2014, pp. 531–540.
[7] H. Liu, Z. Lin, X. Chu, Y.W. Leung, Jump-stay rendezvous algorithm for cognitive radio networks, IEEE Trans. Parallel Distrib. Syst. 23 (10) (2012) 1867–1881.
[8] M.A. L. C. R. Rajan, M.G. Chandra, P. Hiremath, Concepts of graph theory relevant to ad-hoc networks, Int. J. Comput. Commun. Control 3 (2008) 465–469.
[9] N. Meghanathan, Graph theory algorithms for mobile ad hoc networks, Inf. Int. J. Comput. Inf. 36 (2008) 185–199.
[10] Y. Xu, X. Zhong, S. Zhou, Channel assignment in multi-radio wireless ad hoc networks, in: Communications and Networking in China, 2006. ChinaCom '06. First International Conference on, 2006, pp. 1–5.
[11] K. Ramachandran, E. Belding, K. Almeroth, M. Buddhikot, Interference-aware channel assignment in multi-radio wireless mesh networks, in: INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006, pp. 1–12.
[12] C.Z.H. Li, Y. Cheng, P. Wan, Multi-dimensional conflict graph based computing for optimal capacity in mr-mc wireless networks, in: Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, 2010, pp. 774–783.
[13] J. Tang, G. Xue, W. Zhang, Interference-aware topology control and qos routing in multi-channel wireless mesh networks, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2005, pp. 68–77.
[14] W. Li, P. Fan, K.B. Letaief, Joint processing of topology control and channel assignment in wireless ad hoc networks, Wirel. Commun. Mob. Comput. 9 (2009) 269–281.

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003

[15] J. Zhao, H. Zheng, G.-H. Yang, Distributed coordination in dynamic spectrum allocation networks, in: New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on, 2005, pp. 259–268.
[16] V. Raman, N.H. Vaidya, Interference aware channel allocation in a multichannel, multi-interface wireless network, in: Proceedings of the Third ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, 2008, pp. 111–112.
[17] M.X. Gong, S.F. Midkiff, S. Mao, A cross-layer approach to channel assignment in wireless ad hoc networks, Mob. Netw. Appl. 12 (2007) 43–56.
[18] L. Ding, T. Melodia, S. Batalama, M.J. Medley, Rosa: Distributed joint routing and dynamic spectrum allocation in cognitive radio ad hoc networks, in: Proceedings of the 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2009, pp. 13–20.
[19] H.S. Chiu, K. Yeung, K.-S. Lui, J-car An efficient joint channel assignment and routing protocol for ieee 802.11-based multi-channel multi-interface mobile ad hoc networks, Wirel. Commun. IEEE Trans. 8 (2009) 1706–1715.
[20] M. Gong, S. Midkiff, S. Mao, Design principles for distributed channel assignment in wireless ad hoc networks, Commun. ICC IEEE Int. Conf. 5 (2005) 3401–3406.
[21] S.H. Kim, D.W. Kim, Y.J. Suh, A group-based channel assignment protocol for rate separation in ieee 802.11-based multi-radio multi-rate ad hoc networks, Ad Hoc Netw. 10 (1) (2012) 95–110.
[22] H. Cheng, N. Xiong, A.V. Vasilakos, L.T. Yang, G. Chen, X. Zhuang, Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks, Ad Hoc Netw. 10 (5) (2012) 760–773.
[23] Y. Peng, Y. Yu, L. Guo, D. Jiang, Q. Gai, An efficient joint channel assignment and qos routing protocol for ieee 802.11 multi-radio multi-channel wireless mesh networks, J. Netw. Comput. Appl. 36 (2) (2013) 843–857.
[24] D.J. Deng, Y.S. Chen, Y.S. Wong, Adaptive channel allocation strategy for mobile ad hoc networks, Math. Comput. Model. 57 (11) (2013) 2720–2730.
[25] Y. Ding, K. Pongaliur, L. Xiao, Channel allocation and routing in hybrid multichannel multiradio wireless mesh networks, Mobile Comput. IEEE Trans. 12 (2) (2013) 206–218.
[26] Y. Wang, J.J. Garcia-Luna-Aceves, A distributed cross-layer routing protocol with channel assignment in multi-channel MANET, Computing, Networking and Communications (ICNC), International Conference on, IEEE, 2015, pp. 1050–1054.
[27] Y. Ding, L. Xiao, Channel allocation in multi-channel wireless mesh networks, Comput. Commun. 34 (7) (2011) 803–815.
[28] L. Hu, Distributed code assignments for CDMA packet radio networks, in: INFOCOM '91. Proceedings. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking in the 90s., IEEE, 3, 1991, pp. 1500–1509.
[29] A.A. Bertossi, M.A. Bonuccelli, Code assignment for hidden terminal interference avoidance in multihop packet radio networks, IEEE/ACM Trans. Netw. 3 (1995) 441–449.
[30] B.W. Kernighan, L. Shen, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. 49 (2) (1970) 291–307.
[31] K. Andreev, R. Harald, Balanced graph partitioning, Theory Comput. Syst. 39 (6) (2006) 929–939.
[32] Y.Z. Jembre, R. Paul, Y.J. Choi, K.Y. Cheon, C.J. Kim, Channel assignment and jammer mitigation for military MANETs with multiple interfaces and multiple channels, Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, ACM, 2014, pp. 66–72.January
[33] D. Wu, S.H. Yang, L. Bao, C.H. Liu, Joint multi-radio multi-channel assignment, scheduling, and routing in wireless mesh networks, Wirel. Netw. 20 (1) (2014) 11–24.
[34] K.H. Almotairi, X.S. Shen, A distributed multi-channel mac protocol for ad hoc wireless networks, Mobile Comput. IEEE Trans. 14 (1) (2015) 1–13.
[35] J. Jia, J. Chen, J. Yu, X. Wang, Joint topology control and routing for multi-radio multi-channel WMNS under SINR model using bio-inspired techniques, Appl. Soft Comput. 32 (2015) 49–58.
[36] L.H. Yen, K.W. Huang, Link–preserving interference–minimisation channel assignment in multi–radio wireless mesh networks, Int. J. Ad Hoc Ubiquitous Comput. 18 (4) (2015) 222–233.

Please cite this article as: Y.Z. Jembre, Y.-J. Choi, Beacon-based channel assignment and jammer mitigation for MANETs with multiple interfaces and multiple channels, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.003