Chapter 2

OPTIMAL MULTI-CHANNEL ASSIGNMENTS IN VEHICULAR AD-HOC NETWORKS

Hanan Luss and Wai Chen
Telcordia Technologies, Piscataway, New Jersey 08854
hluss@telcordia.com, wchen@research.telcordia.com

Abstract: This paper focuses on establishing a communications path among an ordered sequence of moving nodes, representing vehicles. A channel is used to send information from one node to the next in the sequence on a wireless link. The set of available channels may differ from one node to the next node. Each of the available channels at a node can be used for receiving information from its predecessor node in the sequence or for transmitting information to its successor node in the sequence. However, the same channel cannot be used at a node for both receiving and transmitting information. We present algorithms that determine an optimal sequence of channels that establishes a communications path from the first node to the farthest node possible. We present a depth-first search algorithm that uses a "look-ahead" channel selection rule in order to decrease backtracking. We also present an algorithm that requires only a single pass through the sequence of nodes by identifying optimal channel assignments in subsequences of nodes without a need for backtracking. The latter algorithm requires computational effort that is proportional to the number of nodes in the ordered sequence of nodes.

Keywords: Channel assignments; mobile ad-hoc networks; vehicular ad-hoc networks; wireless communication.

1. INTRODUCTION

A mobile ad-hoc network (MANET) is formed by multiple moving nodes equipped with wireless transceivers. The mobile nodes communicate with each other through multi-hop wireless links, where every node can transmit and receive information. Mobile ad-hoc networks have become increasingly important in areas where deployment of communications infrastructure is difficult. Such networks are used for communications in battle fields, natural disasters, fleets on the ocean, and so forth. Numerous papers have been published on this topic. Representative references include Lin and Gerla (1997), Lin and Liu (1999), and McDonald and Znati (1999). Xu, et al. (2006) describe a framework for multi-channel management in such networks. Earlier related work has focused on channel assignment for traditional packet radio networks with no mobility. Representative studies include Cidon and Sidi (1989), Ephremides and Truong (1990), and Lu (1993).

A vehicular ad-hoc network (VANET) refers to a mobile ad-hoc network designed to provide communications among nearby vehicles and between vehicles and nearby fixed equipment. Chisalita and Shahmehri (2002), Chen and Cai (2005) and Chen, et al. (2006) present a networking approach that uses local peer group architecture in order to establish communications among vehicles. The use of multiple channels allows for simultaneous communication sessions at the logical layer among a network of moving vehicles without partitioning available resources at the physical layer. Thus, the use of multiple channels would significantly increase the network throughput. Existing channel assignment methods use distributed decisions where each vehicle determines which channel to use based on local information on channel availability at neighboring vehicles.

In this paper, we focus on establishing a communications path among an ordered sequence of moving nodes, representing vehicles, using global information of channel availability. The ordered sequence of nodes can be viewed as a directed linear tree topology where a link interconnects a node only to its successor node in the ordered sequence. A channel is used to send information from one node to the next on a wireless link. The set of available channels may differ from one node to the next due to external interferences, other ongoing communications that involve some of these nodes, different equipment used at the nodes, and the like. Each of the available channels at a node can be used for receiving information from its predecessor node in the sequence or for transmitting information to its successor node in the sequence. However, the same channel cannot be used at a node for both receiving and transmitting information. Note that the channel used to transmit information from a node is the channel used to receive information at its successor node in the ordered sequence of nodes. The first node in the sequence, or some nearby system, has as input the information of the set of available channels at each of the nodes in the ordered sequence.

A sequence of channel assignments is called optimal if it establishes a communications path from the first node in the ordered sequence of nodes to the last node in that sequence, or, if such a feasible sequence of channels does not exist, it establishes a communications path from the first node to the farthest node possible. We present two algorithms that determine an optimal sequence of channel assignments. The first algorithm uses a depth-first search starting from the first node in the sequence. We present two versions of the Depth-First Search Algorithm, where the second version improves upon the channel selection rule at a node by using a "look ahead" scheme that may decrease the amount of backtracking, and thus reduce the computational effort. The second algorithm, referred to as the One-Pass Algorithm, requires only a single pass through the sequence of nodes by identifying optimal channel assignments in subsequences of nodes without a need for backtracking, resulting in computational effort that is proportional to the number of nodes in the ordered sequence of nodes.

2. THE MULTI-CHANNEL ASSIGNMENT PROBLEM

Consider a sequence of nodes i, $i = 1, 2, \ldots, N$, where the nodes represent moving vehicles. The ordered sequence of nodes can be viewed as a directed linear tree topology where a link interconnects node i only to node i+1 in the ordered sequence. The sets S_i denote the channels available at node i, $i = 1, 2, \ldots, N$. The number of channels in each of the sets is expected to be quite small, say, between 2 to 4.

Consider the example in Figure 1 with $N = 5$. Each node has a set of channels available for receiving or transmitting information where the term channel is used as a logical entity. It may represent a frequency band (under FDMA), an orthogonal code (under CDMA), and so forth. The set of available channels may differ from one node to the next due to external interferences, other ongoing communications that involve some of these nodes, different equipment used at the nodes, and so forth. For example, node 1 can use channels 1 and 2 as depicted by the set $S_1 = \{1, 2\}$, and node 2 can use channels 1, 2 and 4 as depicted by the set $S_2 = \{1, 2, 4\}$. The information of all sets S_i, $i = 1, 2, 3, 4$ and 5, is provided as input to the channel assignment algorithms. Typically, the input will be available at node 1, however, it may be available at some other location, e.g., at nearby fixed devices. The channel assignment methods are generic and independent of the location of the input.

Each of the available channels at any node i can be used for receiving information from node i-1 or for transmitting information to node i+1. However, the same channel cannot be used at a node i for both receiving and transmitting information. The objective of the channel assignment algorithms is to establish a communications path from node 1 to the farthest node possible in the sequence of ordered nodes. Of course, if possible, the established communications path should reach node N, however, this may not always be possible.

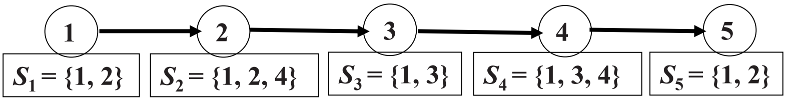

Figure 1. An Ordered Sequence of Nodes and the Available Channels

Let T_i denote the set of available channels at node i and at node i+1, i.e., $T_i = S_i \cap S_{i+1}$. The sets T_i for i = 1, 2, …, N-1 are readily computed from the sets S_i for i = 1, 2, …, N. Thus, in the example of Figure 1, $T_1 = \{1, 2\}$, $T_2 = \{1\}$, $T_3 = \{1, 3\}$, and $T_4 = \{1\}$. Since node i must use on its outgoing link a channel that is in S_i and node i+1 must use on its incoming link a channel that is in S_{i+1}, node i can interconnect with node i+1 only on channels that are in the set T_i. In addition, node i must use different channels for receiving information from node i-1 and for transmitting information to node i+1.

Let f represent index for channels. Let $x_{fi} = 1$ if channel f is selected at node i to interconnect node i to node i+1, and let $x_{fi} = 0$ otherwise. From the discussion above, $x_{fi} = 0$ if $f \notin T_i$. Let $z_i = 1$ if a channel is assigned at node i to interconnect node i to node i+1, and let $z_i = 0$ otherwise. The multi-channel assignment problem is formulated as follows.

The Multi-Channel Assignment Problem

$$Max[\sum_{i=1}^{N-1} z_i] \tag{1}$$

$$z_i = \sum_{f \in T_i} x_{fi}, \quad i = 1, 2, ..., N-1 \tag{2}$$

$$x_{fi} = 0, \quad f \notin T_i, i = 1, 2, ..., N-1 \tag{3}$$

$$x_{fi} + x_{f,i+1} \le 1, \quad f \in T_i \cap T_{i+1}, i = 1, 2, ..., N-1 \tag{4}$$

$$z_{i+1} \le z_i, \quad i = 1, 2, ..., N-1 \tag{5}$$

$$x_{fi}, z_i = 0, 1, \quad f \in T_i, i = 1, 2, ..., N-1. \tag{6}$$

Constraints (2) express decision variables z_i in terms of decision variables x_{fi}, while constraints (3) state that channel f cannot be assigned at node i if $f \notin T_i$. Constraints (4) assure that a node cannot receive and transmit information on the same channel. Constraints (5) state that a channel can be assigned at node i+1 only if a channel is assigned at node i. Constraints (6) define the 0-1 decision variables that are not forced to 0 by constraints (3). Note that by constraints (2) and (6), the number of channels assigned at a node is at most one. Given the constraints above, objective function (1) maximizes the number of assigned channels, which is equivalent to assigning channels to the farthest possible node in the ordered sequence of nodes. Problem (1)-(6) is an integer program with significant structure. A similar formulation without variables z_i can also be provided. In Section 3, we present the Depth-First Search Algorithm for the Multi-Channel Assignment Problem that is not necessarily polynomial as it may require backtracking. In Section 4, we present the One-Pass Algorithm that finds an optimal solution in polynomial time; in particular, for given sets T_i, the computational effort is $O(N)$.

We present below several important observations that will be exploited by the algorithms.

- If each of the sets T_i has two or more channels, a sequence of channels that connects node 1 to node N can readily be assigned. Moreover, this still holds if either T_1 or $T_{N\text{-}1}$ (but not both) have a single channel. Consider an example of five nodes with $T_1 = \{1\}$, $T_2 = \{1, 2\}$, $T_3 = \{1, 2\}$ and $T_4 = \{1, 2\}$. Then, we simply select channel 1 from T_1 and T_3 and channel 2 from T_2 and T_4.
- The challenge is to determine optimal assignments when some of the sets T_i include only one channel. Suppose $T_1 = \{1, 2\}$, $T_2 = \{1, 2\}$, $T_3 = \{1\}$ and $T_4 = \{1, 2\}$. Then, we must select channel 1 from T_1. If channel 2 is selected from T_1, we must select channel 1 from T_2 which, in turn, would prevent selection of channel 1 from T_3.
- Each of the sets T_i can be limited to at most three channels. This is easy to see through the following example. Suppose $T_1 = \{1\}$, $T_2 = \{1, 2, 3\}$ and $T_3 = \{2\}$. Although channels 1 and 2 cannot be selected from T_2, there is always another channel available for selection in T_2.
- If set T_i does not include any channel, a feasible communications path cannot be established beyond node i. Other infeasible cases can also be identified. For example, if sets $T_i = T_{i+1} = \{1\}$, or if sets $T_{i\text{-}1} = \{1\}$, $T_i = \{1, 2\}$ and $T_{i+1} = \{2\}$, a feasible communications path cannot be established beyond node i+1.

3. THE DEPTH-FIRST SEARCH ALGORITHM

The search algorithm constructs a tree, while using backtracking when necessary. We first describe the basic version of the algorithm. From here on we use the notation f_i to denote the channel used for transmitting information from node i to node i+1. Let $FEAS_i$ denote the set of channels that can be used for interconnecting node i with node i+1, given that node i-1 communicates with node i on channel f_{i-1}. Since node i must use different channels for receiving information from node i-1 and for transmitting information to node i+1, $FEAS_1 = T_1$ and $FEAS_i = T_i - f_{i-1}$ for $i > 1$ (obviously, if $f_{i-1} \notin T_i$, then $FEAS_i = T_i$).

Starting from node 1, we select a channel $f_1 \in FEAS_1$. The selection can be done randomly or using any other arbitrary rule. Thereafter, the set T_2 is updated by deleting channel f_1 from T_2 (no update is needed if $f_1 \notin T_2$). The search then continues, attempting to assign a channel that would connect node 2 to node 3. The search continues until a channel successfully interconnects node N-1 to node N, in which case a communications path is established from node 1 to node N, or until at some node $i < N$ $FEAS_i = \emptyset$. In the latter case, backtracking in the search tree is needed. If at some stage $FEAS_1 = \emptyset$, node N cannot be reached for the specified sets S_i, and the search terminates with a communications path that connects node 1 to the farthest node possible.

We illustrate the search algorithm by solving the example shown in Figure 1. We compute the sets $T_i = S_i \cap S_{i+1}$. The resulting search tree is shown in Figure 2.

Starting with node 1, we select a channel from the set T_1 as the candidate channel on the link from node 1 to node 2, using some specified rule, e.g., random selection, the largest or smallest channel index, and so forth. Suppose we select channel 1. Channel 1 is now deleted from the set T_2 since the channel used on the outgoing link from node 2 must differ from the channel used on the input to the node. Since the set T_2 is now empty, there is no available channel to connect node 2 to node 3. So the search on this branch of the tree failed as depicted by in the right part of the tree in Figure 2.

We then backtrack to node 1 and delete channel 1 from set T_1. We now select the remaining channel in T_1, i.e., channel 2, to connect node 1 to node 2 as depicted in the left part of the tree in Figure 2.

Since channel 2 is not in T_2, the set T_2 remains unchanged. Next, we select channel 1 to connect node 2 to node 3. Channel 1 is deleted now from T_3 and we select channel 3 to connect node 3 to node 4. Since channel 3 is not in T_4, the set T_4 remains unchanged. We now select channel 1 to connect node 4 to node 5.

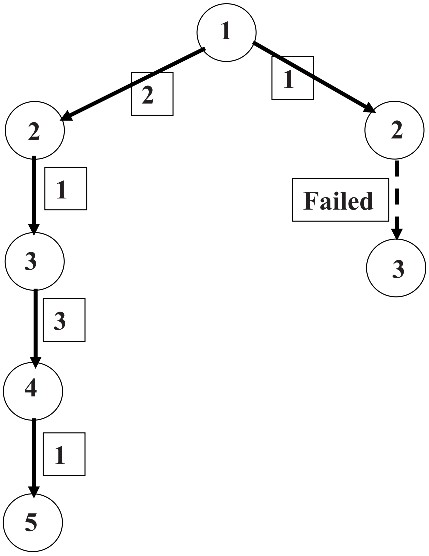

Figure 2. Example of the First-Depth Search Algorithm

We found an optimal sequence of channels that establishes a communications path that connects node 1 to node 5. The path uses channel 2 on the link from node 1 to node 2, channel 1 on the link from node 2 to node 3, channel 3 on the link from node 3 to node 4, and channel 1 on the link from node 4 to node 5.

The search algorithm can be improved by using a "look-ahead" rule to select a channel from among the channels in the set $FEAS_i$. We first examine whether the set $FEAS_i$ includes a channel that is not in the set T_{i+1}, and if so we select such a channel. This selection does not decrease the selection options at node i+1 since it does not affect $FEAS_{i+1}$; hence, it may lead to less backtracking in the search tree. If such a channel does not exist, we use the same selection rule as before.

Figure 3 illustrates the search algorithm by resolving the example shown in Figure 1.

We start by selecting a channel from the set $FEAS_1 = T_1$ as the candidate channel on the link from node 1 to node 2. Since channel 2 is the only channel in $FEAS_1$ that is not in T_2, we select channel 2. By using this "look-ahead" rule we did not select channel 1 which would lead to a failure to establish a connection from node 2 to node 3 as previously demonstrated in

Figure 2. The remaining steps of the search shown in Figure 3 are the same as those shown in the left part of the tree in Figure 2. The search with the "look-ahead" rule found the same optimal solution found by the basic search, but did not require any backtracking.

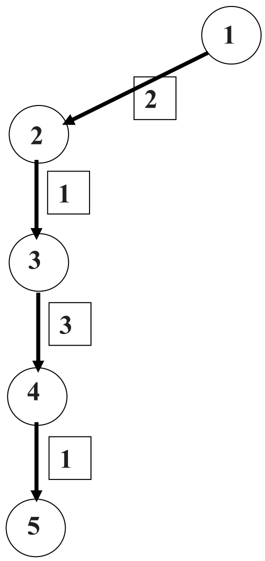

Figure 3. Example of the First-Depth Search Algorithm Using the "Look-Ahead" Rule

Note, however, that the search with the "look-ahead" rule may still require backtracking. This can be easily demonstrated by adding another node to the example of Figure 1, referred to as node 1A, between node 1 and node 2 with the set $S_{1A} = S_1$. Now, since $T_1 = T_{1A} = \{1, 2\}$, starting at node 1, the "look-ahead" rule does not provide any guidance at node 1. If channel 2 is selected at node 1, channel 1 must be selected at node 1A, and no channel would be available to connect node 2 to node 3. Backtracking on the search tree would then be required.

As mentioned in Section 2, we may limit the number of channels in each of the sets T_i to no more than three. Although not guaranteed, this will often reduce the effort spent on backtracking. We should attempt to keep channels that are likely to be selected by the "look ahead" rule; e.g., keep in T_i channels that are not in T_{i+1}.

We now present the search algorithm with the "look-ahead" rule.

The Depth-First Search Algorithm

Initialization

Compute sets $T_i = S_i \cap S_{i+1}$ for $i = 1, 2, \ldots, N$-1.

If T_i has more than three channels, keep only three. If available, keep in T_i channels that are not in T_{i+1}.

$N \leftarrow$ minimum[N, smallest i with $T_i = \varnothing$] (a communications path cannot be established from node 1 to a node beyond the revised N).

Initialize sets $TEMP = BEST = \varnothing$ ($TEMP$ is the interim sequence of channels from node 1 to the currently reached node, and $BEST$ records the longest sequence of channels found since the beginning of the search).

Initialize $i = 1$.

End of initialization.

While $i < N$

$FEAS_i = T_i - f_{i-1}$ (for $i = 1$, $FEAS_i = T_i$).

If $FEAS_1 = \varnothing$, STOP (the set $BEST$ provides the longest possible sequence of channels; at this stopping point the optimal communications path does not reach node N).

If $FEAS_i = \varnothing$ ($i > 1$), backtracking is needed:

Begin

If the sequence of channels in $TEMP$ is longer than that in $BEST$, then update $BEST \leftarrow TEMP$.

Update $T_{i-1} \leftarrow T_{i-1} - f_{i-1}$.

Update $TEMP \leftarrow TEMP - f_{i-1}$.

Update $i \leftarrow i$-1.

Go to beginning of the while loop.

End.

$FEAS_i \neq \varnothing$: Select channel on next link.

Begin

Select channel for transmitting from node i to node i+1 as follows:

If available, select some $f_i \in FEAS_i \backslash T_{i+1}$ (i.e., f_i is in $FEAS_i$ but not in T_{i+1}); otherwise, select some $f_i \in FEAS_i$.

Update $TEMP \leftarrow TEMP + f_i$.

Update $i \leftarrow i$+1.

If $i < N$, go to beginning of the while loop.

End.

End of while loop.

BEST ← *TEMP* (at this stopping point the set *BEST* provides a communications path that connects node 1 to node *N*).

STOP.

End of Depth-First Search Algorithm.

Each of the sets T_i can be obtained by sorting sets S_i and S_{i+1} and merging the sorted sets. Thus, the N-1 sets T_i are computed at an effort of $O(Nhlogh)$ where h is the number of channels in the largest set S_i. Recall that if all sets T_i include two or more channels, a sequence that connects node 1 to node N is obtained through arbitrary selection. The Depth-First Search Algorithm will then find a solution without any backtracking. Nevertheless, in general, the computational effort may grow exponentially with N, where in the worst case the entire search tree will be constructed. The computational effort (after computing the sets T_i and keeping up to three channels in each) is at most $O(3^N)$. We illustrate this through an example in which there is no path that connects node 1 to node N. Let $T_{N\text{-}3} = \{1\}$, $T_{N\text{-}2} = \{1, 2\}$ and $T_{N\text{-}1} = \{2\}$, and let each of the other sets T_i include three channels. Then, the search tree will explore (almost) all possible $O(3^N)$ assignments while failing to find a connection from node N-1 to node N.

4. THE ONE-PASS ALGORITHM

The One-Pass Algorithm does not build a search tree. Instead, it looks for the first node, say node m, along the ordered sequence of nodes that has a single channel in the set T_m and assigns the channel in T_m to the link connecting nodes m to node m+1. The assigned channel is deleted from set $T_{m\text{-}1}$. The algorithm then proceeds backwards to node m-1 and arbitrarily assigns a channel from set $T_{m\text{-}1}$ to the link connecting node m-1 to node m. The algorithm continues in that manner until a channel is assigned to the link connecting node 1 to node 2. Assignment of channels to the links along the path that connects node 1 to node m+1 is completed. Note that the backwards assignment of channels to interconnect the subsequence of nodes 1 to m+1 is guaranteed to succeed since each of the sets T_i for $i = m$-1, m-2, ..., 1 has at least two channels.

If node $m = N$-1, the algorithm terminates since a path is established from node 1 to node N. Suppose $m < N$-1. The assigned channel on the link into node m+1 is deleted from set T_{m+1} and the algorithm searches for the next node in the ordered sequence beyond node m, say node n, that has a single channel in the set T_n. The algorithm then assigns channels to the subsequence of links that connect node n to node n+1, node n-1 to node n, …, node m+1 to node m+2.

The algorithm continues to assign channels to such subsequences until a path that connects node 1 to node N is established or until some updated set, say T_p, is encountered with $T_p = \varnothing$. In the latter case, a communications path can be established only from node 1 to node p.

Note that a subsequence with node N as its last node may have more than one channel in $T_{N\text{-}1}$, in which case the algorithm arbitrarily assigns one of these channels to the link connecting node N-1 to node N.

Figure 4 illustrates the One-Pass Algorithm by resolving the example shown in Figure 1.

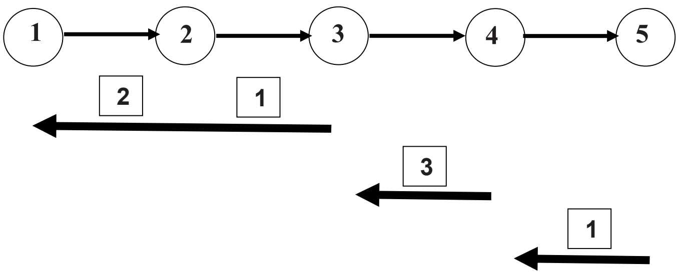

Figure 4. Example of the One-Pass Algorithm

Recall that in Figure 1 S_1 = {1, 2}, S_2 = {1, 2, 4}, S_3 = {1, 3}, S_4 = {1, 3, 4}, and S_5 = {1, 2} resulting in T_1 = {1, 2}, T_2 = {1}, T_3 = {1, 3}, and T_4 = {1}.

Starting from node 1, set T_2 = {1} is the first set with a single channel. Therefore, moving backwards, we select channel 1 on the link from node 2 to node 3, update T_1 by deleting channel 1 from T_1, which results in T_1 = {2}, and select channel 2 on the link from node 1 to node 2. The selection of channels on this subsequence is depicted in Figure 4 by the arrow pointed backwards from node 3 to node 1. Channel 1 is also deleted from set T_3, leading to T_3 = {3}.

Next, starting from node 3, T_3 = {3} is the first set with a single channel. Therefore, the second subsequence includes only a single link and we select channel 3 on the link from node 3 to node 4. The selection of channels on this subsequence is depicted by the arrow pointed backwards from node 4 to node 3. Since T_4 does not include channel 3, no update is needed.

Finally, starting from node 4, node 5 is reached, and channel 1 is selected from node 4 to node 5.

Note that the One-Pass Algorithm found the same solution provided above by either version of the Depth-First Search Algorithm. However, the One-Pass Algorithm finds an optimal sequence in an effort that is proportional to the number of nodes, i.e., in an effort of $O(N)$ (after computing the sets T_i). The sequence found will generate a communications path from node 1 to node N, or, if not possible, from node 1 to the farthest node possible along the ordered sequence of nodes. Consider again the example with $T_{N-3} = \{1\}$, $T_{N-2} = \{1, 2\}$, $T_{N-1} = \{2\}$, where each of the other sets T_i includes three channels. The One-Pass Algorithm will first find a path from node 1 to node N-2, followed by a path from node N-2 to node N-1 (using channel 2). Finally, it will terminate since the updated set T_{N-1} will be empty. The total effort spent on finding a path from node 1 to node N-1 is $O(N)$. On the other hand, as discussed before, the Depth-First Search Algorithm will terminate after spending an effort of $O(3^N)$.

Let $|T_i|$ denote the number of channels in the set T_i. We conclude this section by presenting the One-Pass Algorithm. Note that in this algorithm we do not realize computational savings by deleting channels from the sets T_i that have more than three channels.

The One-Pass Algorithm

Initialization

$T_i = S_i \cap S_{i+1}$ for $i = 1, 2, \ldots, N$-1.

$N \leftarrow$ minimum[N, smallest i with $T_i = \varnothing$] (a communications path cannot be established from node 1 to a node beyond the revised N).

$MIN = 1$.

End of initialization.

Subsequence Channel Assignments

MAX = [i: Smallest $i \geq MIN$ with $|T_i| = 1$]; if no $|T_i| = 1$ set $MAX = N$-1.

Select some $f_{MAX} \in T_{MAX}$.

$i = MAX$.

While $i > MIN$

$T_{i-1} \leftarrow T_{i-1} - f_i$.

Select some $f_{i-1} \in T_i$ - 1.

$i \leftarrow i$-1.

End of while loop.

End of subsequence channel assignments.

Termination Checks

$MIN \leftarrow MAX+1$.

If $MIN = N$, STOP (assigned channels f_1, f_2, …, f_{N-1} provide a communications path from node 1 to node N).

$T_{MIN} \leftarrow T_{MIN} - f_{MIN-1}$.

If $T_{MIN} = \varnothing$, STOP (assigned channels f_1, f_2, …, f_{MIN-1} provide a communications path to the farthest node possible, node MIN).

Go to beginning of Subsequence Channel Assignments.

End of termination checks.

End of One-Pass Algorithm.

5. FINAL REMARKS

We presented two algorithms that determine an optimal sequence of channel assignments that establish a communications path from node 1 to the farthest node possible in a specified ordered sequence of nodes, where the sets of available channels S_i may be node-dependent. For given sets T_i, the One-Pass Algorithm assigns channels in a computational effort *O(N)*, whereas the Depth-First Search Algorithm is not polynomial due to possible backtracking. Hence, for the problem of channel assignments for an ordered sequence of nodes, the One-Pass Algorithm is clearly more efficient. Future research may consist of developing algorithms of multiple channel assignments to more complicated problems where, for example, vehicles may be involved in multiple sessions at the same time. Possible extensions of the algorithms presented in this paper to such problems should be explored. It is an open question which of these algorithms could be effectively extended to more complex problems that cannot be modeled as a single ordered sequence of nodes, but would instead be represented by more general network topologies.

ACKNOWLEDGEMENT

This work was supported by Toyota InfoTechnology Center.

REFERENCES

Chen, W., and Cai, S., 2005, Ad hoc peer-to-peer network architecture for vehicle safety communications, *IEEE Communications Magazine*, April, 100-107.

Chen, W., Cai, S., Chennikara-Varghese, J., Hikita, T., and Lee, J., 2006, Dynamic local peer group organizations for vehicle communications, *Proceedings of Vehicle-to-Vehicle Communications Workshop (V2VCOM 2006) co-located with ACM MobiQuitous*, July 21, San Jose, California.

Chisalita, I., and Shahmehri, N., 2002, A peer-to-peer approach to vehicular communication for the support of traffic safety applications, *Proceedings of the 5th IEEE Conference on Intelligent Transportation Systems*, Singapore, September 3-6, 336-341.

Cidon, I., and Sidi, M., 1989, Distributed assignment algorithms for multi-hop packet radio networks, *IEEE Transactions on Computer* **38:** 1353-1361.

Ephremides, A., and Truong, T., 1990, Scheduling broadcasts in multi-hop radio networks, *IEEE Transactions on Communications* **38:** 456-460.

Lin, C. R., and Gerla, M., 1997, Adaptive clustering for mobile wireless networks, *IEEE Journal on Selected Areas in Communication* **15:** 1265-1275.

Lin, C. R., and Liu, M. J. S., 1999, QoS routing in ad hoc wireless networks, *IEEE Journal on Selected Areas in Communications* **17:** 1426-1438.

Lu, L., 1993, Distributed code assignment for CDMA packet radio networks, *IEEE/ACM Transactions on Networking* **1:** 668-677.

McDonald, A. B., and Znati, T. F., 1999, A mobility-based framework for adaptive clustering in wireless ad-hoc networks, *IEEE Journal on Selected Areas in Communications* **17:** 1466-1487.

Xu, C., Liu, K., Yuan, Y., and Liu, G., 2006, A novel multi-channel based framework for wireless IEEE 802.11 ad hoc networks, *Asian Journal of Information Technology* **5:** 44-47.