An integrated cluster-based multi-channel MAC protocol for mobile ad hoc networks

Zhang, Lili; Soong, Boon Hee; Xiao, Wendong

2007

Zhang, L., Soong, B. H., & Xiao, W. (2007). An Integrated Cluster-Based Multi-Channel MAC Protocol for Mobile Ad Hoc Networks. IEEE Transactions on Wireless Communications, 6(11), 3964-3974.

<https://hdl.handle.net/10356/79956>

<https://doi.org/10.1109/TWC.2007.06038>

© 2007 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <http://dx.doi.org/10.1109/TWC.2007.06038>.

Downloaded on 26 Apr 2023 08:50:20 SGT

An Integrated Cluster-Based Multi-Channel MAC Protocol for Mobile Ad Hoc Networks

^a*Lili Zhang, Member, IEEE, ^aBoon-Hee Soong, Senior Member, IEEE, and
^bWendong Xiao, Member, IEEE*

^a*School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
(e-mail: zhanglili@pmail.ntu.edu.sg; ebhsoong@ntu.edu.sg).*

^b*Institute for Infocomm Research, Singapore (e-mail: wxiao@i2r.a-star.edu.sg).*

Abstract

This paper proposes a novel joint clustering and multi-channel medium access control (MAC) protocol for mobile ad hoc networks, based on a scalable two-phase coding scheme. It employs the first-phase codes for differentiating the clusters and the second-phase codes for distinguishing the nodes in a specific cluster. The proposed protocol effectively integrates the procedure of code assignment with dynamic clustering and henceforth substantially reduces the control overhead of code assignment in a code division multiple access (CDMA) based multi-channel MAC protocol while simultaneously combating the hidden terminal problem. Furthermore, the confliction detection and resolution mechanism for the allocation of the first-phase codes as well as the collision avoidance mechanism for the allocation of the second-phase codes in the control channel are also introduced. Analytical framework and extensive simulation results, in terms of control overhead and delay, are provided and compared to the traditional distributed CDMA based multichannel MAC algorithms with or without clustering.

Index Terms

Mobile ad hoc networks, two-phase coding, multi-channel MAC, clustering, control overhead.

I. INTRODUCTION

Considerable studies have been done in code division multiple access (CDMA) based multi-channel medium access control (MAC) protocol for mobile ad hoc networks (MANETs) [1] [2] [3] [4] [7] [8]. The basic criterion of code assignment for such protocols is that the same codes can only be reused more than two hops away in order to avoid the hidden terminal problem (HTP) [6]. However, all these protocols require that the complete neighborhood knowledge within two-hop separation is known by each node *a priori*. In essence, they assume that nodes broadcast their holding codes to their one-hop and two-hop neighbors such that nodes within two-hop separation can adopt different codes as their transmission codes. The encountered problem is that with the node density and mobility in the network increasing, there will be more collisions in the code broadcasting even with a separated control channel, which cannot ensure the timely acquisition of the accurate neighborhood information of a node. As a consequence, some nodes within two-hop separation may choose the same codes and thus aggravate the HTP. Moreover, frequent neighborhood exchanges of code assignment under high density and mobility will incur a heavy control overhead, which seriously degrades the network performance.

Recently, the multi-cell based location-aware two-phase coding multi-channel MAC protocol (LA-TPCMMP) [9] is proposed for large-scale dense MANETs, where the first-phase codes are used to differentiate cells that are distributed according to hexagonal structure and the second-

phase codes are used to identify nodes in one specific cell. LA-TPCMMP eliminates the HTP without periodical exchanges of neighborhood information and significantly reduces the control overhead. However, its usefulness is partially limited by its dependance on the location information.

In the presence of dynamically changing network topologies due to nodes mobility, clustering concept appears to be an important mechanism employed to build up a hierarchical structure, which may alleviate the heavy control overhead in a large-scale MANET. Through clustering, the topological changes within a cluster do not affect the total network structure and hence mitigate the impact of the mobility on the network performance. However, most of the existing literature focuses only on the pure algorithm design [10][11] and hierarchical topology control [12] as well as clustering-based routing protocol [13], without taking into account the medium access.

Towards the end objective of minimizing the control overhead in the channel assignment of a CDMA-based multi-channel protocol design for a large-scale MANET, we are motivated to develop a integrative protocol by enabling multi-channel MAC protocol with dynamic clustering.

In this paper, we introduce and provide the complete performance analysis of an Integrated Cluster-based Multi-channel MAC Protocol (ICMMP) for a general large-scale MANET. Preliminary idea for this extension has been presented in an earlier version [14]. ICMMP assigns the first-phase codes and second-phase codes respectively for adjacent clusters and different cluster members (*CMs*) in one specific cluster. The salient feature of ICMMP is that code assignment is effectively incorporated with cluster maintenance and thereby significantly reduces the control overhead due to less exchanges of neighborhood information. In ICMMP, each frame is related to a specific first-phase code and each cluster head (*CH*) occupies its particular frame to broadcast its beacon messages for cluster maintenance. Meanwhile, each *CM* in a cluster acquires its unique time slot within the frame of its cluster to broadcast and whenever it is in request for data transmission, it achieves its second-phase code in its cluster using its time slot free of contentions.

The primary contributions of this paper are as follows: 1) present a new protocol that adaptively assigns transmission codes based on the dynamic clustering by utilizing two-phase coding scheme without the requirement of the location information; 2) introduce the confliction detection and resolution mechanism for the first-phase code assignment and present the collision avoidance mechanism of *CMs* during the contention for their respective time slots; 3) provide the theoretical analysis of control overhead and delay with respect to the parameter design; 4) conduct extensive simulations to verify the numerical results.

The rest of the paper is organized as follows. In Section II, we describe the related work. In Section III, we introduce ICMMP, and present the confliction detection and resolution as well as collision avoidance mechanism. In Section IV, we argue its effectiveness of combating the HTP and provide the analytical framework in terms of the control overhead and delay. We describe traditional approaches in Section V. Simulation study and performance evaluation are provided in Section VI, followed by concluding remarks and acknowledgement in Section VII and VIII respectively.

II. RELATED WORK

In the literature, a lot of efforts have been dedicated to multi-channel MAC protocols and clustering algorithms for MANETs. However, these link layer protocols and topology control algorithms are investigated separately. This motivates us to design a joint protocol that combines these two approaches. Below, we discuss several related work in these fields.

A. CDMA-Based Multi-Channel MAC Protocols

Extensive work has been devoted in distributed code assignment in CDMA-based multi-channel MAC protocols to avoid the HTP [1] [2] [4] [7] [8] with the assumption that the complete neighborhood knowledge within two-hop separation is known by each node *a priori* via broadcasting.

Various distributed code assignment schemes are discussed in [1], including transmitter-oriented code assignment (TOCA), receiver-oriented code assignment (ROCA) and pairwise-oriented code assignment (POCA). The optimal code assignment schemes for networks with special topologies and heuristic distributed code assignment algorithms are proposed in [2] with the aim to minimize the number of orthogonal codes and eliminate the hidden terminal interference. In [4], the communication complexity of distributed assignment of codes in the worst case is given, by considering the case that a node determines its transmission code through broadcasting to all neighbors within two-hop separation. In SEEDEx algorithm [7], nodes exchange the seeds of their pseudo-random number generators within two-hop separation to know the schedules of each other such that the hidden terminals and exposed terminals can be identified in advance and reduced. A novel distributed neighbor-aware contention resolution algorithm HAMA is introduced in [8] by defining different priorities to nodes and determining the channel access schedule for each time slot. However, HAMA requires prompt detection and notifying the change within two-hop separation to harmonize the code scheduling to avoid the HTP.

B. Access Based Clustering Algorithms

Recently, the access based clustering algorithms have received some attention [15][16][17][19]. For example, the access based clustering protocols ABCP [15] seeks to minimize the overhead on cluster formation by considering broadcast scheduling of control messages and CABP [16] uses the randomized control channel broadcast access method with the purpose of maximizing the worst-case control channel efficiency. More closely related to our work is the study on providing mobility-aware and access-based clustering in MANETs. Typically, the scheme [17] formulates and realizes the virtual transmitter or receiver clusters onto the total scenario by packing the appropriate transmitter-receiver pairs to reduce the multiple access interference [18] for a CDMA-based spread spectrum physical layer technology. It shows that explicitly inducing spatial clustering in contention has substantial promise towards achieving high spatial reuse in spread spectrum ad hoc networks. However, it does not address the critical problem of the channel assignment in such a cluster induced CDMA-based multi-channel MAC protocol. To the best of our knowledge, so far, only MAPLE clustering scheme [19] points to the possibility of taking the medium access of adjacent clusters into account during the cluster establishment and maintenance, where the CHs of adjacent clusters respectively occupy their particular time frames to broadcast their beacon messages and maintain their clusters. However, it does not explain how to allocate multi-channel resources by making use of clustering information to broaden the

efficient channel spatial reuse.

III. ICMMP

A. Protocol Preliminaries

1) *Two-Phase Coding Scheme*: In ICMMP, the first-phase codes are used for differentiating the adjacent clusters and the second-phase codes are employed for distinguishing the nodes in a specific cluster. Let us assume there exist a set of pseudo noise (PN) orthogonal codes for data transmission. We divide it into m sub-sets with n codes in each sub-set. The first-phase code is actually the sequence number of the sub-sets and the second-phase code is the sequence number of the PN orthogonal codes in one sub-set. Let C_1^i denote the i_{th} sub-set and C_2^j denote the j_{th} PN code in any particular sub-set. Then a node with its transmission code denoted as (C_1^i, C_2^j) indicates that its transmission code is the j_{th} PN code in the i_{th} sub-set. In this way of denotation of a joint first-phase and second-phase code, a node can identify an actual PN orthogonal code as its transmission code for its data transmission.

Let $S(\mathbf{i})$ denote the set of nodes of cluster $\mathbf{i}$ and $C_1(\mathbf{i})$ denote the first-phase code of cluster $\mathbf{i}$. Then for any node $a \in S(\mathbf{i})$ with its first-phase code and second-phase code denoted as $C_1(a)$ and $C_2(a)$ respectively, we have its transmission code $C(a) = (C_1(a), C_2(a))$ and $C_1(a) = C_1(\mathbf{i})$. We call cluster $\mathbf{i}$ and cluster $\mathbf{j}$ ($\mathbf{i} \neq \mathbf{j}$) are neighboring clusters if there exists $u \in S(\mathbf{i})$ and $v \in S(\mathbf{j})$ such that u and v are one-hop away from each other. The set of neighboring clusters of cluster $\mathbf{i}$ is denoted as $\mathcal{NC}(\mathbf{i})$.

2) *Code Confliction and Criteria for Code Assignment*: We call two nodes within two-hop separation are in code confliction when they adopt the same transmission codes since collision may happen at their common neighbors if they transmit data packets simultaneously. In order to avoid the confliction, we introduce the following criteria for code assignment in ICMMP.

Criterion 1: For any two nodes a and b in a given cluster $\mathbf{i}$, their second-phase codes should be assigned such that $C_2(a) \neq C_2(b)$.

Criterion 2: For any two neighboring clusters $\mathbf{i}$ and $\mathbf{j}$, their first-phase codes should be assigned such that $C_1(\mathbf{i}) \neq C_1(\mathbf{j})$.

Criterion 3: For any two clusters $\mathbf{i}$ and $\mathbf{j}$, if there exists a cluster $\mathbf{k}$ that satisfies cluster $\mathbf{i} \in \mathcal{NC}(\mathbf{k})$ and cluster $\mathbf{j} \in \mathcal{NC}(\mathbf{k})$, their first-phase codes should be assigned such that $C_1(\mathbf{i}) \neq C_1(\mathbf{j})$.

Considering the above criteria for code assignment, a cluster $\mathbf{i}$ maintains a set of *Permitted First-phase Codes* $C_{PFC}(\mathbf{i})$ which consists of all the first-phase codes that are different from those of its neighboring clusters.

B. Protocol Operation Mechanism

We assume that there exist three different states for nodes, respectively namely *CM*, *CH* and cluster candidate, where a *CH* is responsible for maintaining and assigning the second-phase codes to its *CM*s in each cluster and each *CM* is one-hop away from its *CH*. The wireless channel is divided into a control channel and a data channel. The network is synchronized and control channel operate in time-slotted mode. The identification of transmission code is completed in the

control channel and the data transmission that is based on CDMA technique is done in the data channel.

1) *Initial Clustering Algorithm*: The selection of the first-phase codes and second-phase codes can be based on various kinds of initial clustering algorithms. Here, without loss of generality, suppose the nodes initially employ the random competition-based clustering (RCC) [12] to establish the clusters, for cluster maintenance in ICMMP. That is, any node that does not belong to any cluster can initiate a cluster formation by broadcasting a declaration message to claim itself as a *CH* after deferring a random timer, and the first candidate node that broadcast will be elected as the *CH* by its neighbors. All its neighbor nodes, upon hearing such a broadcast, give up their right to be a *CH* and become members of this cluster. During the cluster maintenance, if two *CH*s are within the communication range of each other, the *CH* with lower ID will give up its leadership and notify its one-hop neighbors not within the overlapped area to re-compete for the *CH*, in order to reduce the overlapping of the clusters. Unavoidably, there exists overlapping between clusters. A node within the overlapped area will choose the *CH* with the best receiving signal strength and thereby join the corresponding cluster finally.

2) *Access Format of Control Channel and Basic Idea*: In this subsection, we give the access format of control channel and basic idea of ICMMP. As shown in Fig. 1. a, the control channel is time-slotted and divided into super-frames and frames. Each super-frame consists of M_1 frames, wherein each frame is dedicated for one cluster with M_1 being the available number of frames that can be used for the adjacent clusters.

Each frame is composed of one beacon slot (*BS*) followed by some fixed number of M_2 maintenance slots (*MS*s), wherein the *BS* is dedicated to election, probing as well as beacon messages of a *CH* and the $MS_l (1 \leq l \leq M_2)$ is dedicated for a *CM* within a cluster with M_2 being the supported number of *CM*s in a cluster. Each *MS* is further composed of three mini-slots, where the first two mini-slots are dedicated to Request for Slot (RFS) message of a *CM* and Confirmation of Slot (COS) message of its *CH* respectively and the third mini-slot is for broadcasting message of the *CM*. There is a flag that indicates code request and code release in the RFS message and broadcasting message respectively.

The basic idea of ICMMP is that each frame is related to a corresponding first-phase code. In the control channel, different frames of a super-frame are dedicated to adjacent clusters. Thus clusters that are spatially separated enough can use the same frame. A *CH* broadcasts its beacon messages in the *BS* of its particular frame for cluster maintenance, which simultaneously broadcasts its adopted first-phase code. Correspondingly, in the data channel, adjacent clusters adopt the different first-phase codes and clusters that are spatially separated can adopt the same first-phase codes.

Meanwhile, in the control channel, each *CM* in a cluster acquires its unique *MS* within the frame of its cluster to broadcast messages for the purpose of negotiating the appropriate frames for its cluster without confliction with each other. Whenever it is in request for data transmission, it achieves its second-phase code in its cluster using its *MS* free of contentions.

Hence, the first-phase and second-phase code acquisition for a node is embedded into the frame acquisition and slot acquisition respectively. This effectively integrates code assignment

with clustering. Moreover, in this way, there exists no collisions from broadcasting message of the adjacent clusters in the control channel by the regulated frame access. We explain below how to select the first-phase and second-phase codes.

3) *Selection of The First-Phase Codes via Conflicition Detection and Resolution*: The selection of the first-phase code for one specific cluster is determined by the first-phase codes held by its neighboring clusters. The first-phase code of a cluster is acquired through negotiating the appropriate frame during the cluster maintenance and henceforth we recite how to obtain the first-phase code for a cluster that occupies the corresponding frame in the control channel. Note that each *CM* keeps a stored table that records the first-phase codes held by their neighboring clusters.

CH election is completed in the *BS*. Whenever a cluster candidate successfully contends to be the *CH* of a cluster $\mathbf{i}$, it listens for a duration of a super-frame, waits until the beginning of a new free frame and sends the *probing message* that disseminates its presumed first-phase code in the *BS*. The *CMs* of cluster $\mathbf{i}$ that receive the probing message check their stored tables, judge whether the confliction occurs as well as inform confliction by sending the *broadcasting messages* that indicates whether the neighboring clusters are adopting the same first-phase code. The stored table of each *CM* is updated according to the information overheard from the broadcasting message of the *CMs* of its neighboring clusters. Upon collecting the broadcasting messages of its *CMs*, the *CH* updates its $C_{PFC}(\mathbf{i})$ and randomly chooses one free first-phase code from its $C_{PFC}(\mathbf{i})$ to continue to probe its appropriate first-phase code which should not be in confliction with any neighboring clusters to satisfy Criterion 2. The procedure of the negotiation of the first-phase codes is termed as confliction detection and confliction resolution. We call the probing of the first-phase code with the probability $\frac{1}{|C_{PFC}(\mathbf{i})|}$ as the probability-based confliction resolution scheme.

The *CH* that does not receive the opposition messages, considers the current first-phase code to be suitable and periodically broadcasts the *beacon message* that disseminates its cluster ID (determined by its *CH* ID), first-phase code and the number of its *CMs*. Otherwise, it should adjust its first-phase code such that Criterion 2 can be satisfied. The *CMs*, which hear the beacon messages, will record the corresponding first-phase code information.

The *CMs* send their broadcasting messages that convey the neighboring cluster ID and their first-phase codes as well as the number of their *CMs*. Thus, a *CH* may detect that its neighboring clusters with different cluster ID occupy the same frame and adopt the same first-phase code, and thereby do the corresponding adjustment to ensure Criterion 3. We, therefore, introduce another criterion for the adjustment,

Criterion 4: For any two clusters $\mathbf{i}$ and $\mathbf{j}$ that satisfy $\mathbf{i} \in \mathcal{NC}(\mathbf{j})$ or there exists a cluster $\mathbf{k}$ that satisfies cluster $\mathbf{i} \in \mathcal{NC}(\mathbf{k})$ and cluster $\mathbf{j} \in \mathcal{NC}(\mathbf{k})$, if $C_1(\mathbf{i}) = C_1(\mathbf{j})$ and $|S(\mathbf{i})| < |S(\mathbf{j})|$, we assume cluster $\mathbf{j}$ keeps the original first-phase code and cluster $\mathbf{i}$ re-obtains its first-phase code according to the probability-based confliction resolution scheme.

Through this kind of approach, the first-phase code confliction between clusters can be resolved within a limited number of trials as will be proved in Section IV.

4) *Selection of The Second-Phase Codes*: Whenever a cluster is established, a *CH* maintains its available slots set (ASS) in its cluster. Meanwhile, the *CMs* that have joined this cluster may contend to obtain their respective *MSs* in the specific frame of this cluster. Since it is difficult for all *CMs* to coordinate their response and not collide with each other in each frame, we employ binary splitting strategy [9][22] to resolve collisions during the slot acquisition, i.e., when there exist the collisions from RFS messages during contention, the senders that can not hear COS message will decide with equal probability $p = 1/2$ whether it will continue to transmit RFS in the next *MS*. A *CH* will send COS after a successful reception of RFS and meanwhile update its ASS so that other nodes involved in the collision can restart the contention following this strategy. A *CM* can obtain its slot in the frame of its cluster within a limited number of trials as will be proved in Section IV. New *CMs* will always listen for a period of their respective frame to know the ASS from the beacon message before participating in the contention for their *MSs*.

As shown in Fig. 1. b, a *CM* that has obtained its slot can acquire its second-phase code from its *CH* free of contention by marking the code request field of RFS message sent in its allocated *MS* whenever in request for data transmission. Upon receiving the allocated code from COS, the *CM* acknowledges its *CH* by sending broadcasting message so that its *CH* can update the recorded available second-phase codes for its cluster. Meanwhile, the broadcasting message conveys its destination ID such that the destination node can achieve code synchronization when overhearing the code information.

5) *Data Transmission*: Transmitter-oriented data transmission [5] is adopted. The data transmission may be initiated immediately in the data channel after the acquisition of transmission code. As soon as the data flow transmission finishes, the *CM* will return its second-phase code to its *CH* by marking the code release field in its periodically transmitted broadcasting message. Moreover, when a *CM* migrates into another cluster, it releases its original second-phase code to the former *CH* and re-obtains its new second-phase code in the new cluster. As a complementary measure, the *CH* will recycle the allocated second-phase codes to facilitate the reutilization of codes if it can not receive the broadcast that indicates the use of the code within a predefined interval.

C. Discussion

Overall, ICMMP provides us a useful framework of jointly designing clustering and channel assignment in a multichannel MAC protocol, which is of significance in reducing the control overhead and henceforth energy consumption. Here, we concentrate on the CDMA-based multi-channel MAC protocol due to the fact that heavy control overhead is incurred in the code assignment of the traditional CDMA-based multi-channel protocol. However, based on the point of channel spatial reuse, the proposed model in the present paper is also applicable to time division multiple access (TDMA) and frequency division multiple access (FDMA) based multi-channel MAC protocol as well as their combinations, although with their own uniqueness. For example, in the form of FDMA, each frame in ICMMP can be related to a specific frequency sub-band and thus in the data channel the neighboring clusters can use different frequency sub-bands for data transmission to avoid the collisions between data packets from *CMs* within neighboring clusters. Within one cluster, when each *CH* occupies its particular frame in the control channel to broadcast its beacon messages for cluster maintenance, it may schedule and assign the different time slots in the form of TDMA (or different codes in the form of CDMA) to

the different members for the data transmission in the data channel. The clusters that are spatially separated can use the same frame in the control channel, i.e., use the same frequency sub-bands in the data channel. In this view of point of channel spatial reuse, the proposed model is applicable to TDMA, FDMA based multi-channel MAC protocol as well as their combinations.

ICMMP is presented based on the background that multi-codes system is available according to the existing work [23][24]. It pays the cost of not minimizing the required quasi-orthogonal codes compared to identifying a different code within two-hop separation via neighborhood exchanges. Nevertheless, this tradeoff may be acceptable in the context of MANETs, as reducing control overhead is of vital importance in some specific applications that are strictly restricted by power supply.

IV. THEORETICAL ANALYSIS

A. Effectiveness in Combating HTP

We first demonstrate that ICMMP effectively combats HTP during data transmission. Considering that there may exist overlapping among the clusters, we argue the cases that the HTP may happen through the following Lemma,

Lemma 1: In ICMMP, any two nodes A and B that are in different clusters and within two-hop separation belong to two clusters which are either neighboring clusters or neighbors of another common cluster.

Proof: Let R denote the transmission range and $D(a, b)$ denote the distance between nodes a and b respectively. If $D(a, b) \leq R$, $a \in S(1)$ and $b \in S(2)$, then cluster $1 \in \mathcal{NC}(2)$. Otherwise, assume $D(a, c) \leq R$ and $D(b, c) \leq R$, i.e., node c is the common one-hop neighbor of nodes a and b . If $a \in S(1)$, $b \in S(2)$ and $c \in S(1) \cup S(2)$, then cluster $1 \in \mathcal{NC}(2)$. Whereas, if $a \in S(1)$, $b \in S(2)$ and $c \in S(3) \in \bar{S}(1) \cup \bar{S}(2)$, we still have cluster $3 \in \mathcal{NC}(1)$ and cluster $3 \in \mathcal{NC}(2)$ since $D(a, c) \leq R$ and $D(b, c) \leq R$. Consequently, nodes a and b belong to two clusters which are either neighboring clusters or neighbors of another common cluster.

Next, according to Lemma 1, the total cases that HTP may happen can be illustrated in Fig. 2. Suppose the transmission codes of nodes a and b are $C(a) = (C_1(a), C_2(a))$ and $C(b) = (C_1(b), C_2(b))$ respectively. As shown in Fig. 2 (a), when a and b are in the same cluster 1, then $C_1(a) = C_1(b)$ and $C_2(a) \neq C_2(b)$ according to Criterion 1, thus $C(a) \neq C(b)$. In Fig. 2 (b), when a and b are in the adjacent clusters 1 and 2 respectively, then $C_1(a) \neq C_1(b)$ according to Criterion 2. In Fig. 2 (c), when a and b are respectively in cluster 1 and 3 that are the neighbors of cluster 2, then $C_1(a) \neq C_1(b)$ according to Criterion 3. Thereby, no matter what second-phase codes nodes a and b use in the second and third cases, one always has $C(a) \neq C(b)$. In all three cases, $C(a)$ and $C(b)$ are always quasi-orthogonal. Therefore, if three criteria are always observed, there is no HTP in ICMMP.

B. Control Overhead and Delay Analysis

Next, we provide the analytical framework of ICMMP in terms of control overhead and delay. Let us assume that nodes are spatially distributed in the network according to Poisson process with density ρ . Then, we have $N = \rho\pi R^2$ being the average number of nodes within the

transmission range R . Note that the initial cluster establishment only requires one successful broadcasting *per* cluster. As an initial phase, it does not contribute significantly to the overall control overhead that is further composed of the periodical *CH* beacon and *CM* broadcasting for the cluster maintenance. Since the acquisition of the first-phase code and second-phase code are respectively incorporated into the frame acquisition and slot acquisition during cluster maintenance. Next, the derivation of the average incurred control overhead and delay in the frame acquisition and slot acquisition is shown by referring to the control channel access that has been described in detail in Section III.

1) *Control Overhead and Delay in the Slot Acquisition*: The successful slot acquisition requires the correct reception of the RFS and COS messages. However, it is in the presence of the collisions during the reception of RFS messages, since nodes within the communication range of a *CH* may initiate their contentions for their respective *MS*s at the same time.

Initially, among N nodes, the probability of k contenders follows a Poisson distribution $P_1(k) = e^{-N} N^k / k!$. Given ζ_k as the average number of contentions required to resolve one collision of k contenders, we have the expression of ζ_k that follows the recursive relationship

$$\zeta_k = \begin{cases} 1, & k = 1 \\ \frac{1 + 2^{-k} \sum_{j=1}^{k-1} \binom{k}{j} \zeta_j}{1 - 2^{-k+1}} & k > 1 \end{cases} \quad (1)$$

according to the conclusion of binary splitting strategy in [22]. Accordingly, the average number of the RFS messages to win one contention χ_1 can be easily computed as

$$\chi_1 = \sum_{k=1}^{\infty} \frac{P_1(k)}{k} \sum_{l=1}^k l \sum_{j=0}^{\zeta_l-1} \left(\frac{1}{2}\right)^j. \quad (2)$$

From here on, unless otherwise mentioned, without loss of generality, we assume $\rho = 10^{-4}$ in the following analysis. The numerical values of χ_1 for the increasing transmission ranges 100m, 150m, 200m and 250m are respectively 3.30659, 7.18795, 12.6519 and 19.6893. Although it is seen that the upward slope of χ_1 increases with the incremental transmission ranges, we would like to show that ICMMP is less sensitive to the increasing value of transmission ranges than TAs in Section VI.

Recall that the *CM*s contend to obtain their *MS*s of their cluster only in their specific frames. Given χ_2 as the average total number of frames required for k nodes that participate in the contention to obtain their respective slots, we have

$$\chi_2 = \frac{1}{M_2} \sum_{k=1}^{\infty} \sum_{j=1}^k \zeta_j P_1(k) \quad (3)$$

since a frame is composed of M_2 *MS*s. Fig. 3 shows χ_2 varies with the transmission ranges and

M_2 . We observe that although χ_2 increases with a larger transmission ranges given a certain M_2 , it quickly decreases with the incremental value of M_2 under a certain transmission range. This implies that one can adjust the value of M_2 for a certain R to satisfy a certain delay requirement.

In steady state, the number of nodes simultaneously participating in the contention can be computed according to the number of nodes that migrate into the communication range of a CH within the interval of a super-frame. Let v denote the average speed and $N_{m_h1}(\varepsilon)$ denote the total number of nodes which migrate into the one-hop distance of a CH in the pre-specified small time interval ε . Note that $v\varepsilon$ is far smaller than R which can be easily shown later. Therefore, given l as the distance between two nodes, the average number of new coming nodes can be approximately derived as

$$N_{m_h1}(\varepsilon) = \int_R^{R+v\varepsilon} \theta \rho(2\pi l) dl \quad (4)$$

where $\theta = \frac{2 \arccos((l-R)/(v\varepsilon))}{2\pi}$ denotes the angle along which nodes can migrate into the required area.

Denote the duration to transmit a RFS, a COS, a beacon message from a CH and broadcasting message from a CM respectively as τ_r , τ_c , τ_{bh} and τ_{bm} seconds. In the following, unless otherwise stated, we assume $\tau_{bh} = \tau_{bm}$, $\tau_r = \tau_c$ and the length of the broadcasting message is 50 bytes with $\tau_{bm} = 0.00133s$. Since a node sends RFS message which disseminates only its ID and a bit which indicates if the second-phase code is required or not and a CH sends COS message that conveys the sequence number of the allocated slot or code, the duration of these mini-slot is less than that of mini-slot for broadcasting message. We define a variable $\kappa = \frac{\tau_r}{\tau_{bm}}$. Let τ_{sf} and τ_f denote the duration of a super-frame and a frame respectively, then we have $\tau_{sf} = M_1 \tau_f$ and $\tau_f = \tau_{bh} + M_2(\tau_r + \tau_c + \tau_{bm}) = \tau_{bh} + M_2(1 + 2\kappa)\tau_{bm}$. When M_1 adopts the maximal value 20, $M_2 \doteq \rho\pi(250)^2$ and $\kappa = 1/5$, we have $\tau_{sf} = 0.7714s$. Usually, the average speed is less than $20m/s$. Hence, we have $v\varepsilon = v\tau_{sf} \ll R$ when $R \geq 100m$.

As shown in Fig. 4, we observe $N_{m_h1}(\tau_{sf})$ versus v and R_d . As the transmission range and speed increase, $N_{m_h1}(\tau_{sf})$ increases quickly. However, when $R \leq 250m$, we observe that the number of nodes migrating into the transmission range of a CH within τ_{sf} is less than 1. Hence, in the steady status, it takes less contention for a new coming node to obtain its MS .

Since the CH acknowledge the CM s only when it wins the contention, it adds 1 to the overall overhead due to the successful contention. Thus, the average number ϕ_S of required RFS and COS messages for a successful slot acquisition of a node is

$$\phi_S = \chi_1 + 1 \quad (5)$$

where the additional 1 is due to one COS message after a successful contention from RFS messages.

Since CMs of a cluster only send control packets for slot acquisition in a specific frame of a super-frame, the average total delay of ψ_S spent in the appropriate slot acquisition can be approximately derived as

$$\psi_S = \lfloor \chi_2 \rfloor \times \tau_{sf} + \frac{\tau_{sf}}{2} + \tau_{bh} + (\chi_2 - \lfloor \chi_2 \rfloor) \times (1 + 2\kappa) \times \tau_{bm}. \quad (6)$$

As shown in Fig. 5, we observe the relationship of average incurred control overhead and average total delay versus $1/\kappa$. Both the average control overhead and average total delay decrease with the declined value of κ , which means that one can reduce the control overhead and delay by adjusting the duration of the mini-slots.

2) *Control Overhead and Delay in the Frame Acquisition:* Let us assume that there are averagely x clusters probing their first-phase codes and y free frames are available within the neighborhood of any given cluster (including this cluster). Now we proceed to compute the average number of required control packets for a successful first-phase code broadcast of a node and the average total number of super-frames required to resolve a confliction.

Let $\alpha_{k,i}$ denote the number of clusters which have chosen the i_{th} frame in the k_{th} probing and $\nu_{k,i}$ denote the number of times that different number repeatedly appears in $\alpha_{k,1}, \dots, \alpha_{k,y}$ in turn, then we have $\alpha_{k,i} = 0, 1, 2, \dots, x$ and $\nu_{k,i} = 1, 2, \dots, y$ for any $1 \leq i \leq y$. According to the classical occupancy problem [21], the probability to obtain the given occupancy number $\alpha_{k,1}, \dots, \alpha_{k,y}$ equals

$$\mathcal{P}_2(x, y, k) = \frac{x!}{\alpha_{k,1}! \alpha_{k,2}! \dots \alpha_{k,i}! \dots \alpha_{k,y}!} \times \frac{y!}{\nu_{k,1}! \nu_{k,2}! \dots \nu_{k,i}! \dots \nu_{k,y}!} \times y^{-x} \quad (7)$$

where $\sum_{i=1}^y \alpha_{k,i} = x$ for any $1 \leq k \leq y_0$. Here, y_0 is the pre-specified maximal number of probing after which there is no confliction in the frame acquisition of the neighboring clusters.

Here, we use the Dirac function with $\delta(0) = 1$ and $\delta(z) = 0$ if $x \neq 0$. Let α_k and β_k denote the number of the clusters whose first-phase codes should be re-acquired in the k_{th} probe and the number of the available frames that have not been occupied by the neighboring clusters respectively, then we can easily have

$$\alpha_{k+1} = \sum_{i=1}^y (1 - \delta(\alpha_{k,i} - 1) - \delta(\alpha_{k,i})) (\alpha_{k,i} - 1) \quad (8)$$

and

$$\beta_{k+1} = \beta_k - \sum_{i=1}^{\beta_k} (1 - \delta(\alpha_{k,i})) \quad (9)$$

where $\alpha_1 = x$, $\beta_1 = y$ and $1 \leq k \leq y_0$.

Let ϑ_k denote the probability that there is the confliction that two clusters choose the same frame in the k_{th} probe. Since confliction occurs only when $\alpha_{k,i} \neq 1$ or 0, according to the conditional probability, we have

$$\vartheta_{k+1} = \delta\left(\prod_{i=1}^y (\delta(\alpha_{k,i} - 1) + \delta(\alpha_{k,i}))\right) \vartheta_k \mathcal{P}_2(\alpha_k, \beta_k, k) \quad (10)$$

for $1 \leq k \leq y_0$ with $\vartheta_1 = 1$.

Thus, according to the probability of any possible occupancy combination, the average number of required control packets of ϕ_F for a successful first-phase code broadcast of a node is

$$\begin{aligned} \phi_F &= \chi_3 \\ &= \sum_{t=1}^{y_0} \sum_{s=1}^t \sum_{\sum_{i=1}^y \alpha_{s,i} = \alpha_s} \alpha_s \prod_{k=1}^t \mathcal{P}_2(\alpha_k, \beta_k, k) \\ &\quad \times \delta(\vartheta_{t+1}). \end{aligned} \quad (11)$$

Fig. 6 shows χ_3 varies with x and y . We observe that χ_3 increases with the incremental x and decreases with the incremental y . This indicates that an incremental number of the available frames is expected to reduce the control overhead for a successful first-phase code broadcast without confliction.

Similarly, the average total number of super-frames of χ_4 required to resolve a confliction and satisfy Criteria 2-3 is

$$\chi_4 = \sum_{t=1}^{y_0} \sum_{s=1}^t \sum_{\sum_{i=1}^y \alpha_{s,i} = \alpha_s} \prod_{k=1}^t \mathcal{P}_2(\alpha_k, \beta_k, k) t \delta(\vartheta_{t+1}). \quad (12)$$

As shown in Fig. 7, we find that χ_4 decreases quickly with the increasing y given a certain x . This tell us that a larger number of available frames may reduce the delay for a successful first-phase code broadcast without confliction with each other.

The average delay of ψ_F for a *CH* to obtain its appropriate frame is upper bounded by

$$\psi_F = \chi_4 \tau_{sf} / x. \quad (13)$$

3) *Average Total Control Overhead and Delay*: Let P_{CH} and P_{CM} denote the probability of a node to be a *CH* and a *CM* respectively, then we approximately have

$$P_{CH} = \frac{1}{N} \quad (14)$$

and

$$P_{CM} = \frac{N-1}{N}. \quad (15)$$

Define the average lifetime of a *CH* or *CM*, i.e., T_{CH} and T_{CM} , as the average duration of a node to be a *CH* or *CM* within a cluster continuously. Instead of average total delay, we get the average delay ψ_S for a node in slot acquisition, with χ_2 replaced with $\chi'_2 = \frac{1}{M_2} \sum_{k=1}^{\infty} \sum_{j=1}^k \zeta_j \frac{P_1(k)}{k}$ in Eq.(6). Note that the total control packets incurred in the average lifetime of a *CH* or *CM* are composed of the control packets of code acquisition and those of periodical broadcasting. Therefore, the control overhead ϕ_{CA} of ICMMP in unit time is upper bounded by

$$\phi_{CA} = P_{CH} \frac{\phi_F + \lceil \frac{T_{CH} - \psi_F}{\tau_{sf}} \rceil}{T_{CH}} + P_{CM} \frac{K\phi_S + 1 + \lceil \frac{T_{CM} - \psi_S}{\tau_{sf}} \rceil}{T_{CM}} \quad (16)$$

where the additional 1 in the second component is due to one broadcasting message subsequent to the successful slot acquisition.

Similarly, the average delay ψ_{CA} of code acquisition is upper bounded by

$$\psi_{CA} = P_{CH} \frac{\psi_F}{T_{CH}} + P_{CM} \frac{\psi_F + \psi_S}{T_{CM}}. \quad (17)$$

C. The Selection of Parameters

Let N_c denote the maximal number of neighboring clusters, then y should be an appropriate value that satisfy the delay requirement during the negotiation and adjustment of the first-phase codes. In the worst case, $x = N_c + 1$ clusters are contending for their appropriate first-phase codes and $y = M_1$ frames are available. Then, given the pre-specified average delay requirement d_1 , we can determine the appropriate value of M_1 according to $\psi_F \leq d_1$ by referring to Eq.(12) and Eq.(13). Under a certain κ and M_1 , we require the suitable value of M_2 which satisfy $\psi_S \leq d_2$, where d_2 is the required average maximal delay for a node to obtain its slot.

As M_2 implies the supported number of *CMs* in a cluster, in our simulations, we assume that M_2 follows the normal distribution and determine M_2 as $\mu + \sigma$, where μ and σ are respectively the mean value and standard deviation of the number of *CMs* within the communication range of a *CH*. As an example, the chosen value M_2 under the transmission range 100m and the ensuing delay of slot acquisition are 4.7893 and 1.67675, respectively.

V. TRADITIONAL ALGORITHMS (TAS)

In TAS, code assignment requires the exchanges of two-hop neighborhood information. Based on the criterion that nodes need to know the information of nodes within two-hop separation, the algorithm in [4] gives the communication complexity of distributed assignment of codes in the worst case as $O(N_n d_m^2)$, where N_n is the total number of nodes and d_m is the network degree. We simulate the TAS without clustering according to [4], i.e., a node sends a code assignment

message (CAM) to all its one-hop neighbors when a new node comes into transmission range and all receivers are required to acknowledge the sender to ensure the reliable transmission of CAM. In TAs with clustering, the same code assignment algorithm is employed with the exception that the clustering mechanism described in Section III.B is underlying. That is, each node follows the control channel access mode of the regulated clustering maintenance and sends its CAM only in the third mini-slot of its *MS* rather than sending randomly, for the purpose of neighborhood broadcasting.

VI. PERFORMANCE EVALUATION

In this section, we present the numerical and simulation results of ICMMP. The simulation is done in Ns2 [20]. The performance is evaluated in terms of control overhead and delay versus transmission range and average speed. We choose the topology as $1000m \times 1000m$ with 100 nodes randomly distributed and vary the transmission range R to get the different number of one-hop neighbors. We use two-ray ground model as the propagation model, AODV as the routing protocol as well as the random way-point mobility model [25] with average speed up to $20m/s$ and 0 pause time. We choose the random way-point mobility model to illustrate performance of ICMMP, since as a stochastic model to describe the movement behavior of mobile nodes it is the most commonly used mobility model in simulation studies of ad hoc network protocols and may approximate the random walk mobility model with the large input parameter [25]. The channel rate is $2Mbps$ with control channel rate of $0.3Mbps$. The traffic type is CBR UDP with packet size $512bytes$. Assume all the control packets are of the same length $50bytes$. The simulation time is $500s$ and the simulation results are averaged 10 runs with different movement patterns for each value of traffic arrival rate of λ , ν and R . The parameters are set to be $\kappa = 1/5$, $M_1 = 20$ and $x = 8$.

Fig. 8 shows average control overhead with 95% confidence interval versus the transmission ranges when $\lambda = 50Pks/s$ and $\nu = 10m/s$. When the network is considerably less dense with $N \doteq 3$ ($R = 100m$), the control overhead of TAs without clustering is even less than that of ICMMP. This is due to that when the network is loose, the neighborhood update of TAs is less frequent and the probability of collisions is small. Whereas, in ICMMP, extra control overhead is caused by cluster establishment and maintenance with the appropriate frame and slot acquisition, which makes it less advantageous. However, ICMMP significantly outperforms TAs when the network is becoming dense. When $N \doteq 20$ with $R = 250m$, the control overhead incurred in TAs with clustering is almost 11 times higher than that of ICMMP. Furthermore, the control overhead incurred in TAs without clustering is almost 23 times higher than that of ICMMP. Moreover, the confidence interval of ICMMP is much smaller than those of TAs with clustering and without clustering. Thus, one may draw the conclusion that control overhead of ICMMP is not affected significantly by density since the acquisition of the first-phase and second-phase codes is completely incorporated in the clustering maintenance. Although TAs with clustering show a superior performance than TAs without clustering with the increasing number of one-hop neighbors, we observe that the control overhead in both algorithm of TAs is obviously more than that of ICMMP and aggravates seriously with the increasing transmission ranges no matter it is based on flat topology or clustering.

Fig. 9 shows average control overhead with 95% confidence interval versus average speeds when $R = 200m$ and $\lambda = 50Pks/s$. With the speed increasing, there is a higher chance for a new

node to migrate into transmission range, which requires more updating of neighborhood information in TAs. Consequently, the control overhead of TAs under both topology almost aggravates linearly with the increasing speed. ICMMP also requires more control packets in response to increasing speed since high mobility requires more cluster maintenance. However, the upward trend of TAs is significantly worse and the confidence interval is much larger than those of ICMMP. Therefore, we have that ICMMP substantially reduces control overhead for a sufficiently dense or high mobility network. As shown in Table I, the simulation results of ICMMP match well with the numerical results, which validates our analysis. Here, note that the control overhead first decreases with the incremental transmission ranges and then increase slightly with the incremental transmission ranges. This is attributed to the fact that the average lifetime of the *CH* and *CM* both increases with the incremental transmission ranges.

Fig. 10 shows how average delay varies with the transmission ranges under the varying average speed. It is observed that average delay of both ICMMP and TAs with clustering is higher than that without clustering as with clustering a node has to wait for its appropriate slot in the appropriate frame of its cluster to transmit its neighborhood information. However, the lower delay of TAs without clustering is obtained at the cost of higher overhead from frequent updating of neighborhood information. Moreover, we observe that average delay degrades quickly with the increasing transmission ranges. In contrast, the upward slope in ICMMP and TAs with clustering is decreasing. In ICMMP, with the increasing speed, there are more *CM*s to migrate into a new cluster, which causes more collisions in the appropriate slot assignment for the acquisition of the second-phase code. Meanwhile, the first-phase codes of the adjacent clusters have to be adjusted timely to avoid the HTP. As a consequence, the delay of ICMMP increases. Nevertheless, ICMMP incurs less delay and rises less sharply than TAs with clustering due to less exchanges of control packets. Table II shows the simulation results of ICMMP are close to the numerical results.

VII. CONCLUDING REMARKS AND FUTURE WORK

This paper proposes a novel multi-channel MAC protocol for MANETs termed as ICMMP, which integrates a scalable two-phase coding scheme with dynamic clustering. For this protocol, to decide an appropriate confliction-free transmission code, the complete two-hop neighborhood information of a node is not necessary, which makes it immune from serious overhead that other TAs suffer from. Although one-hop neighborhood of clusters is indispensable, relative to the code assignment of nodes within two-hop separation, the probability of selecting an appropriate first-phase code that is not in confliction with neighboring clusters greatly increases since the impact of code identification of clusters suffering from the increasing density and mobility is much less than that incurred by nodes within two-hop separation. By detection and resolution of code confliction, ICMMP can combat the HTP during data transmission. The reuse of the first-phase codes greatly increases the scalability of ICMMP. The analysis with respect to the control overhead and delay are derived and validated by extensive simulation results. It is demonstrated that ICMMP substantially improves the system performance in terms of control overhead, throughput and energy efficiency.

In the future, one can investigate the overall system performance by combining ICMMP with some more advanced clustering algorithms such as CBRP [13]. The energy issue, such as

transmission power control for mitigation of multiple access interference [18], in ICMMP can also be studied.

VIII. ACKNOWLEDGEMENT

The authors would like to acknowledge all the anonymous reviewers for their valuable comments that have helped to improve this paper.

REFERENCES

- [1] L. Hu, "Distributed code assignment for CDMA packet radio networks," *IEEE/ACM Trans. Networks*, vol. 1, no. 67, pp. 668–677, Dec. 1993.
- [2] A. A. Bertossi and M. A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop radio networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 441–449, Aug. 1995.
- [3] A. Ephremides and T. V. Truong, "Scheduling broadcast in multihop radio networks," *IEEE Trans. Commun.*, vol. 38, pp. 456–460, Apr. 1990.
- [4] J. J. Garcia, "Distributed assignment of codes for multihop packet-radio networks," in *Proc. IEEE MILCOM*, Nov. 1997.
- [5] T. Makansi, "Transmitter-oriented code assignment for multi-hop radio networks," *IEEE Trans. Commun.*, vol. 35, no. 12, 1987.
- [6] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: the hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Commun.*, vol. 23, pp. 1417–1433, 1975.
- [7] R. Rozovsky and P. R. Kumar, "SEEDEX: a MAC protocol for ad hoc networks," in *Proc. MobiHoc*, Oct. 2001.
- [8] L. Bao and J. J. Garcia-Luna-Aceves, "Distributed transmission scheduling using code-division channelization," in *Proc. IFIP-TC6 Networking*, May 2002.
- [9] L. L. Zhang, B. H. Soong, and W. D. Xiao, "Location-aware two-phase coding multi-channel MAC protocol (LA-TPCMMP) for MANETs," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1659–1669, May 2007.
- [10] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sept. 1997.
- [11] M. Gerla and J. T. -C. Tsai, "Multicluster, mobile, multimedia radio network," *ACM-Baltzer J. Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.
- [12] K. X. Xu, X. Y. Hong, and M. Gerla, "An ad hoc network with mobile backbones," in *Proc. IEEE ICC*, Apr. 2002.
- [13] M. Jiang, J. Li, and Y. C. Tay, "Cluster based routing protocol (CBRP)," draft-ietf-manet-cbrp-spec-01.txt, Aug. 1999.
- [14] L. L. Zhang, B. H. Soong, and W. D. Xiao, "Cluster-adaptive two-phase coding multi-channel MAC protocol for MANETs," in *Proc. IEEE ICC*, May 2005.
- [15] T. C. Hou and T. J. Tsai, "An access-based clustering protocol for multihop wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 7, pp. 1201–1210, July 2001.
- [16] Z. Cai, M. Lu, and X. Wang, "Channel access-based self-organized clustering in ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 2, no. 2, April-June 2003.
- [17] X. Y. Yang and G. D. Veciana, "Inducing spatial clustering in MAC contention for spread spectrum ad hoc networks," in *Proc. ACM MobiHoc*, May 2005.
- [18] A. Muqattash and M. Krunz, "CDMA-based MAC protocol for wireless ad hoc networks," in *Proc. ACM MobiHoc*, June 2003.
- [19] R. Palit and E. Hossain, "Mobility-aware access-based clustering in mobile wireless ad hoc networks," thesis, University of Manitoba, 2004.
- [20] The CMU Monarch project's wireless and mobility extensions to NS2, Aug. 1999. Available at: <http://www.monarch.cs.cmu.edu>
- [21] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, third edition. New York: John Wiley & Sons, Chapter 2.

- [22] M. Zaozi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Trans. Mobile Computing*, vol. 2, no. 4, Oct.–Dec. 2003.
- [23] F. H. P. Fitzek, "Quality of service support in wireless multi-code CDMA systems," Ph. D thesis, Technical University Berlin, Telecommunication Networks Group TKN, 2002.
- [24] M. Krishnam, M. Reisslein, and F. Fitzek, "Analytical framework for simultaneous MAC packet transmission (SMPT) in a multicode CDMA wireless system, *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 223–242, Jan. 2004.
- [25] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. and Mobile Computing*, vol. 2, pp. 483–502, 2002.

Lili Zhang received her B.S. degree in electronic engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China in 1999, the M.S. degree in communication engineering from China Academy of Telecommunication Technologies (CATT), Beijing, China in 2002, and the Ph.D. degree in communication engineering from school of electrical and electronic engineering, Nanyang Technological University (NTU), Singapore in 2006. She is presently with the Intelligent System Center, NTU, as a research fellow. Her current research interests involve multi-channel medium access control design, topology control, energy efficient resource management and cross-layer design for mobile ad hoc networks and wireless sensor networks.

Boon-Hee Soong received his B.Eng. (Hons I) degree in Electrical and Electronic Engineering from University of Auckland, New Zealand and the Ph.D. degree from the University of Newcastle, Australia, in 1984 and 1990 respectively. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University. From October 1999 to April 2000, he was a Visiting Research Fellow at the Department of Electrical and Electronic Engineering, Imperial College, UK under the Commonwealth Fellowship Award. He also served as a consultant for Mobile IP in a recent technical field trial of Next-Generation Wireless LAN initiated by IDA (InfoComm Development Authority, Singapore). He is holder to Tan Chin Tuan Fellowship award with an attachment at Duke University in June 2004. He has served as Technical Programme Committee for several conferences Globecom, VTC and ISWCS. He has been teaching a number of subjects related to the field of Network Performance and wireless communications. His area of research interests includes Sensor Networks, Mobile Ad-hoc networks, mobility issues, optimization of wireless networks, routing algorithms, system theory, Quality of service issues in high-speed networks, and signal processing. He has published over 100 international journals and conferences. He is currently a Senior member of IEEE and a member of ACM.

Wendong Xiao received the B.S. degree in Mathematics, and the Ph.D. degree in Automatic Control from the Northeast University, China, in 1990 and 1995, respectively. From April 1996 to April 1999, he worked at the POSCO Technical Research Laboratories, Pohang, South Korea as a Post-Doctor Researcher. From May 1999 to December 2001, he was with the Northeastern University, China as an Associate Professor. From February 2001 to July 2004, he worked at the Nanyang Technological University, Singapore as a Research Fellow. Since July 2004, he has been with the Institute for Infocomm Research (I2R), Singapore as a Scientist and a Senior Research Fellow. His current research interests include collaborative signal processing, information fusion, MAC and routing protocols, Quality of Service (QoS), and cross-layer design in wireless ad hoc and sensor networks.

List of Tables

TABLE I	COMPARISON OF RESULTS OF AVERAGE CONTROL OVERHEAD VERSUS TRANSMISSION RANGE AND SPEED IN ICMMP
TABLE II	COMPARISON OF RESULTS OF AVERAGE DELAY VERSUS TRANSMISSION RANGE AND SPEED IN ICMMP

List of Figures

- Fig. 1 Control channel access format.
- Fig. 2 Diagram of HTP in ICMMP.
- Fig. 3 χ_2 versus transmission range and M_2 .
- Fig. 4 Average number of nodes migrating into the transmission range of a CH within the duration of a super-frame versus transmission range under the varying average speeds.
- Fig. 5 Average control overhead and average total delay of slot acquisition versus $1/\kappa$.
- Fig. 6 χ_3 versus x and y (plot only $y \geq x$).
- Fig. 7 χ_4 versus x and y .
- Fig. 8 Average control overhead versus transmission range.
- Fig. 9 Average control overhead versus average speed.
- Fig. 10 Average delay versus transmission range under the varying average speeds

Table I (A)				
R(m) ($v = 10m/s$)	100	150	200	250
Simulation results	1.9613	0.7393	0.5885	0.6327
Numerical upper bound	2.1593	0.8996	0.7171	0.7801
Table I (B)				
v(m/s) ($R = 200m$)	5	10	15	20
Simulation results	0.3850	0.5885	0.7837	0.9663
Numerical upper bound	0.4970	0.7171	0.9310	1.1458

TABLE I

R(m)	100	150	200	250
Simulation results(v=10m/s)	0.0354	0.0320	0.0570	0.0730
Numerical upper bound(v=10m/s)	0.0530	0.0451	0.0695	0.0871
Simulation results(v=20m/s)	0.0541	0.0494	0.0903	0.1151
Numerical upper bound(v=20m/s)	0.0730	0.0621	0.1032	0.1302

TABLE II

(a) Control channel access format

(b) Acquisition procedure of the second-phase code

Fig. 1

Fig. 2

Fig. 3

Fig. 4

Fig. 5

Fig. 6

Fig. 7

Fig. 8

Fig. 9

Fig. 10