<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Topology-Transparent Scheduling in Mobile Ad Hoc Networks With Multiple Packet Reception Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Liu, Y; Li, VOK; Leung, KC; Zhang, L</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>IEEE Transactions on Wireless Communications, 2014, v. 13, p. 5940-5953</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>2014</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/217038">http://hdl.handle.net/10722/217038</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; ©2014 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.</td>
</tr>
</tbody>
</table>
Topooy-Transparent Scheduling in Mobile Ad Hoc Networks With Multiple Packet Reception Capability

Yiming Liu, Member, IEEE, Victor O. K. Li, Fellow, IEEE, Ka-Cheong Leung, Member, IEEE, and Lin Zhang, Member, IEEE

Abstract—Recent advances in the physical layer have enabled wireless devices to have multiple packet reception (MPR) capability, which is the capability of decoding more than one packet, simultaneously, when concurrent transmissions occur. In this paper, we focus on the interaction between the MPR physical layer and the medium access control (MAC) layer. Some random access MAC protocols have been proposed to improve the network performance by exploiting the powerful MPR capability. However, there are very few investigations on the schedule-based MAC protocols. We propose a novel \(m\)-MPR-\(l\)-TTS algorithm for mobile ad hoc networks with MPR, where \(m\) indicates the maximum number of concurrent transmissions being decoded, and \(l\) is the number of codes assigned to each user. Our algorithm can take full advantage of the MPR capability to improve the network performance. The minimum guaranteed throughput and average throughput of our algorithm are studied analytically. The improvement of our \((m, l)\)-TTS algorithm over the conventional topology-transparent scheduling algorithms with the collision-based reception model is linear with \(m\). The simulation results show that our proposed algorithm performs better than slotted ALOHA as well.

Index Terms—Medium access control (MAC), multiple packet reception (MPR), topology-transparent scheduling (TTS).

I. INTRODUCTION

T

RADITIONALLY, medium access control (MAC) algorithms assume the collision-based reception model, in which a packet failure (equivalently collision) happens, if more than one interfering neighbor of a user transmit simultaneously. Thus, the fundamental goal of the conventional MAC algorithms is to improve the throughput by resolving collisions. In random access MAC protocols, each user is allowed to transmit probabilistically with the goal of minimizing the collision probability. In schedule-based MAC protocols, each user is scheduled to transmit in order to avoid collisions. However, this collision-based reception model does not reflect the capabilities of the advanced wireless transceivers nowadays. Advanced signal processing technologies have enabled wireless transceivers to correctly decode multiple packets transmitted simultaneously, referred to as the multiple packet reception (MPR) capability. Various technologies can be used to enable the MPR capability, and can be categorized into two different groups, namely, training-based algorithms and so-called blind interference cancellation algorithms.

In training-based algorithms such as space-time coding and multiuser detection (MUD) with successive interference cancellation (SIC) [20], the knowledge of propagation channels and signal waveforms is assumed to be available. However, this is difficult to obtain and may be impractical in mobile ad hoc networks. In blind interference cancellation algorithms [11], [22], [34], the signal of interest is modulated by a known amplitude or phase variation. This allows the intended receiver to estimate and suppress the interfering sources without knowing the channel states. These blind interference cancellation algorithms can separate the colliding packets and handle the near-far effect [22] without assuming the knowledge of channel, rendering them suitable for mobile ad hoc networks. However, they require a higher computational complexity in signal processing during the packet decoding phase. Yet, the complexity is acceptable. It has been shown in [22] that the complexity of the blind interference cancellation algorithm increases linearly with the increasing \(m\), where \(m\) is the maximum number of concurrent transmissions being decoded. Thus, we assume that the underlying MPR capability in this paper is achieved by this kind of blind interference cancellation algorithms.

In this paper, we focus on studying the cross layer interaction between the MPR physical and MAC layers, rather than the technologies that enable the MPR capability. Specifically, we focus on the throughput in the MAC layer, defined as the number of successful transmission. Under the \(m\)-MPR model, in which the receiver can correctly decode at most \(m\) packets simultaneously, we propose an \(m\)-MPR-\(l\)-code topology-transparent scheduling \((m, l)\)-TTS algorithm. Time is divided into equal-sized time slots, grouped into frames. In each frame, a user is allocated some transmission slots based on the assigned codes. In the traditional topology-transparent scheduling algorithms with the collision-based reception model, each user is assigned a unique code to achieve the maximum throughput.

Manuscript received July 11, 2013; revised January 9, 2014 and May 28, 2014; accepted September 6, 2014. Date of publication September 17, 2014; accepted September 6, 2014. Date of publication September 17, 2014;

© 2014IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
The powerful MPR capability allows us to assign more codes to a user, since a collision does not necessarily lead to a packet failure with the MPR capability. In our \((m, l)\)-TTS algorithm, we assign \(l\) transmission codes to each user due to the fact that the increase in the number of packet failures is much less than that in the transmission time slots, resulting from the \(m\)-MPR capability. We believe that the difference between the proposed algorithm and the traditional topology-transparent scheduling algorithms is nontrivial. Firstly, in our algorithm, the optimal frame structure that maximizes the throughput is totally different from that in the traditional algorithms. Secondly, assigning multiple codes (polynomials) to a user introduces a new problem, namely, we have to assign multiple codes (polynomials) in such a way to avoid coincidences among different codes (polynomials).\(^1\) We detail the assignment of multiple codes in Section III. In addition, we show that the improvement of our \((m, l)\)-TTS algorithm with the \(m\)-MPR capability over the conventional topology-transparent scheduling algorithms with the collision-based reception model is linear with increasing \(m\). The simulation results show that our proposed algorithm performs better than random access protocols, namely, slotted ALOHA with MPR.

A. Motivation and Related Work

Several random access MAC protocols have been proposed \([16, 17, 21]\) to improve the network performance by taking advantage of the powerful MPR capability. The random access approaches cannot provide deterministic delay and throughput bounds. That is why most networks offering throughput and delay guarantees, say, the tactical networks, such as the Enhanced Position Location Reporting System (EPLRS) \([19]\) and the Joint Tactical Radio System (JTRS) \([26]\), implement deterministic, schedule-based protocols, such as time division multiple access (TDMA).

The related work on schedule-based protocols can be categorized into two different groups, namely, topology-dependent and topology-transparent, based on whether the detailed network connectivity information is required. Existing topology-dependent approaches focus on finding a minimum-length, conflict-free schedule based on the detailed network topology. This problem is proved to be NP-complete \([2, 13, 23]\). More importantly, re-computation and information exchanges are required to maintain accurate network topology information and distribute the new schedules when the network topology changes. Distributed topology-dependent scheduling algorithms are introduced in \([25]\). However, it has been shown that it takes several minutes to obtain and distribute the updated schedules \([25]\), which is too long in mobile networks, especially for real-time traffic. Thus, the robustness and effectiveness of these topology-dependent scheduling algorithms are undermined in large, highly dynamic, wireless mobile ad hoc networks.

In order to overcome the aforementioned limitations in mobile ad hoc networks, topology-transparent scheduling algorithms have been proposed \([5, 7, 14, 18]\). In these algorithms, no updates on network connectivity information are required. In topology-transparent scheduling algorithms, each node is assigned multiple time slots for transmission in each frame. It is guaranteed that there is at least one conflict-free slot per frame. Chlamtac and Farago \([7]\) developed a topology-transparent algorithm that guarantees at least one collision-free time slot in each frame time, but the performance is even worse than the conventional TDMA in some cases. Ju and Li \([18]\) proposed another algorithm to maximize the minimum guaranteed throughput. However, only unicast communication is considered. Cai et al. \([5]\) proposed a broadcast scheduling algorithm, modified Galois field design (MGD), which sends the same message multiple times during one frame time in order to guarantee exactly one successful broadcast transmission per frame.

However, there are few schedule-based MAC algorithms with MPR, such as TDMA. The conventional TDMA protocols \([2, 9, 13, 23, 25]\) construct transmission schedules to avoid collisions and maximize the throughput. This does not match the characteristics of MPR as it allows collisions. We believe that this is the inherent reason why there is much less interest in the idea of designing schedule-based TDMA protocols with MPR. However, we find that topology-transparent scheduling algorithms \([5, 7, 14, 18]\) are suitable for the MPR capability. In the conventional topology-transparent scheduling algorithms with collision-based reception, each user is assigned a unique transmission schedule or code. Although the transmission schedules are not collision-free, each user is guaranteed to have at least one conflict-free transmission slot per frame.

Most existing MAC protocols with the MPR capability are random access algorithms. Slotted ALOHA was extensively studied in \([16, 17]\) and can significantly improve network performance. A multiple access protocol based on receiver-controlled transmission was proposed in \([21]\). Although this work pointed out that it is necessary to redesign the MAC protocol so as to make the best use of the MPR capability, it can only be used in predefined topologies and cannot be extended to an arbitrary ad hoc network. Another random access protocol was proposed in \([33]\). It takes advantage of the MPR capability to support users with different quality of service requirements. Some work \([6, 8, 33]\) focused on carrier sense multiple access (CSMA) with the MPR capability. However, it has been shown in \([8]\) that the throughput gain of CSMA over slotted ALOHA is very limited, especially when the MPR capability increases. All these CSMA algorithms are designed only for wireless local area networks (WLANs) and cannot be applied in multihop ad hoc networks, since it has been shown that CSMA suffers from serious instability and unfairness issues in such networks \([31]\). Reference \([28]\) studied the effect of the MPR capability on the network capacity and energy efficiency. A survey paper on MPR for wireless random access networks can be found in \([20]\). Reference \([27]\) proposed a polynomial-time heuristic algorithm to solve the joint routing and scheduling problem with the MPR capability. It is similar to our work, as its scheduling algorithm can provide throughput and delay guarantees. However, it is a centralized algorithm which needs to know all the detailed network connectivity and traffic information, making it impractical in mobile networks.

\(^1\) A coincidence of two polynomials is defined in Section II-B.
In this paper, we propose a schedule-based scheduling algorithm, which is oblivious to the network changes in mobile networks and takes advantage of the powerful MPR capability. Thus, the proposed algorithm can provide throughput and delay guarantees in mobile networks with the MPR capability.

The remainder of this paper is organized as follows. In Section II, we present our system model and some definitions and theorems which will be used in the following sections. Section III presents the details of our proposed algorithm. The average throughput and packet delay of the proposed algorithm are also derived. Extensive simulations are conducted to evaluate the performance of our proposed algorithm in Section IV. The effect of inaccuracies in the estimation of network parameters such as the maximum node degree and the number of network nodes on the performance of the proposed algorithm is also investigated. We conclude in Section V.

II. System Model

A. Network Model

A mobile ad hoc network consisting of \(N\) nodes can be represented by a graph \(G(V,E)\). \(V\) is the set of all network nodes labeled from \(1\) to \(N\) and \(E\) is the set of all edges. If Node \(v\) is within the interference range of Node \(u\), an edge denoted by \((u,v)\) is in \(E\). We assume that if \((u,v)\) \(\in E\), \((v,u)\) \(\in E\). The degree of a node \(u\), \(D(u)\), is defined as the number of its interference neighbors. The maximum node degree \(D_{\text{max}}\) is defined as \(D_{\text{max}} = \max_{u \in V} D(u)\). We assume that \(D_{\text{max}}\) is much smaller than the number of nodes \(N\) and remains constant while the network topology changes [9]. In practice, empirical data may be used to estimate \(D_{\text{max}}\). In addition, \(D_{\text{max}}\) is always pessimistically estimated to ensure that the actual number of interfering neighbors does not exceed the estimate. We show that the performance of our algorithm is insensitive to the accuracy in the estimation of \(D_{\text{max}}\) in Section IV. Thus, it is unnecessary to use distributed online method to obtain the maximum node degree when the network operates.

We focus on TDMA networks. Time is divided into equal-sized frames. Each frame is further divided into \(p\) subframes, each of which consists of \(p\) synchronized, equal-sized time slots, where \(p\) is a prime. The frame structure is shown in Fig. 1. Fig. 1 demonstrates some transmission time slots of an arbitrary node \(i\), in which an “A” is located. The transmission time slots are determined by one of its time slot allocation functions, which is defined and discussed later. Synchronization can be achieved by Global Positioning System (GPS) or the synchronization algorithms such as the one proposed in [12].

The MPR capability can be modeled by an \(n \times n\) MPR matrix \(C\) [16], [17], [21], [27], [33], where \(C_{n,i}\) is the conditional probability that \(i\) packets are correctly decoded given that \(n\) packets are transmitted simultaneously by the nodes in the interference range of a receiving node.

The MPR matrix is a function of many parameters, such as the channel conditions, the modulation technologies, and the signal processing technologies, and thus can take many different forms. In this paper, we assume a generally used strong \(m\)-MPR model [16], [17], [21], [27], [33]. In this case, a user can correctly decode all the packets if the number of concurrent transmissions within its interference range is not greater than \(m\), and none of the packets if the number of concurrent transmissions exceeds \(m\). That is,

\[
C_{i,i} = 1,
\]

if \(i = 1, 2, \ldots, m\), and

\[
C_{n,i} = 0,
\]

otherwise. The conventional collision-based reception model can be modeled by the 1-MPR matrix. We also investigate another \(m\)-MPR model achieved by the blind interference cancellation algorithm using polynomial phase-modulating sequences [22] in Section IV.

We assume that the transmission channel is error-free. Thus, the transmission from Node \(u\) to Node \(v\) fails when: 1) Node \(v\) is also transmitting, or 2) there are more than \(m - 1\) other nodes in \(v\)’s interference range transmitting simultaneously. Note that the transmission from \(u\) to \(v\) fails when \(v\) is also transmitting. This is because the self-interference of \(v\) is much stronger that the received signal from \(u\), leading to the fact that \(v\) with the MPR capability cannot decode a packet from \(u\). This problem can be solved by applying different full-duplex technologies [4]. However, it is beyond the scope of this work.

B. Definitions and Theorems

In the following, we present some definitions and theorems used throughout the paper.

**Definition 1** [18]: A polynomial with degree \(k \mod p\) can be expressed as \(f(x) = \sum_{i=0}^{k} a_i x^i (\mod p)\).

**Definition 2**: A coincidence of two polynomials with degree \(k \mod p\) is defined as the root of the difference of these two polynomials. That is, if \(f_u(j) - f_v(j) = 0\), \(j\) is the coincidence of \(f_u(x)\) and \(f_v(x)\), where \(j = 0, 1, \ldots, p - 1\).

**Theorem 1** [18]: There are at most \(k\) coincidences of two arbitrary different polynomials with degree \(k \mod p\).

**Proof**: The proof can be found in [18].

**Definition 3**: For a given network \(G(V,E)\), each node \(v\) is assigned a time slot allocation function (TSAF) group consisting of \(l\) unique TSAFs, i.e., polynomials with degree \(k \mod p\), namely, \(\{f_j^v(x)\}\), where \(j = 1, 2, \ldots, l\). The polynomials are assigned to each node in such a way that any two polynomials of an arbitrary node has no coincidence. The detailed method of assigning polynomials is discussed in the next section.

This TSAF group is used to calculate the allocated transmission time slots in one frame for each node. For an arbitrary node \(v\), it transmits in \(l\) time slots \(\{f_j^v(i)\}\) in Subframe \(i\),
where $i = 0, 1, \ldots, p - 1$ and $j = 1, 2, \ldots, l$. Thus, each node transmits in $lp$ time slots in one frame. We refer to these $lp$ allocated transmission time slots of an arbitrary node $v$ as its time slot location vector (TSLV), $TSLV(v)$, i.e., $TSLV(v) = (f^1_v(0), \ldots, f^j_v(0), f^1_v(1), \ldots, f^j_v(1), \ldots, f^l_v(p - 1), \ldots, f^l_v(p - 1))$.

**Theorem 2:** There are at most $kl^2$ coincidences of two arbitrary nodes $u$ and $v$, each of which is assigned $l$ different polynomials with degree $k$ mod $p$.

**Proof:** According to Theorem 1, there are at most $k$ coincidences of an arbitrary polynomial of Node $v$ and an arbitrary polynomial of Node $v$. Each node has $l$ unique polynomials. Thus, there are at most $kl^2$ coincidences of two arbitrary nodes $u$ and $v$. $\blacksquare$

**Definition 4:** Given the $m$-MPR model, the number of packet failures for a transmission from an arbitrary node $v$ to its destination $u$, $N_f$, is defined as the number of assigned transmission slots as specified in $TSLV(v)$, in which Node $u$ also transmits or there are more than $m - 1$ other neighbors of Node $u$ other than $v$ transmitting.

Before proceeding, we use Figs. 2, 3, and Table I to illustrate the aforementioned definitions of $l$, polynomial, coincidence, and packet failure in detail. Considering the network shown in Fig. 2, Node 1 is transmitting to Node 2 and Nodes 3, 4, 5 are the interfering neighbors of Node 2. We assume that $m = 2$, $k = 1$, $p = 5$, and $l = 2$. That is, each node is assigned two polynomials with degree $1$ mod $p = 5$, as listed in Table I. Accordingly, the transmission time slots of each node are shown in Fig. 3. Consider the transmission from Node 1 to Node 2. A “C” in Subframe $i$ ($i = 0, 1, \ldots, 4$) of Node 1 indicates that $i$ is a coincidence of Node 1’s and other nodes’ polynomials. Similarly, an “F” in Subframe $i$ ($i = 0, 1, \ldots, 4$) of Node 1 indicates a packet failure for a transmission from Node 1 to Node 2. A packet failure for the transmission from Node 1 to Node 2 happens in Subframe $i$ if $i$ is a coincidence of Node 1’s and Node 2’s polynomials or more than $m - 1$ polynomials assigned to Nodes 3, 4, and 5 have the same coincidence $i$ with those of Node 1’s polynomials.$^2$

**Theorem 3:** Given that the $m$-MPR model and each node is assigned $l$ unique polynomials with degree $k$, $N_f \leq N_f^{max} = \min(lp, S)$, where

$$S = kl^2 + \left[\frac{(D_{max} - 1)kl^2}{m}\right].$$

**Proof:** Each node has at most $D_{max}$ neighbors. Thus, $l$ polynomials of an arbitrary node $v$ has at most $(D_{max} - 1)kl^2$ coincidences with the $l(D_{max} - 1)$ polynomials of the other $D_{max} - 1$ interfering neighbors of its destination $u$. Recall that if there are more than $m - 1$ out of these $D_{max} - 1$ interfering neighbors of $u$ also transmitting, the transmission from $v$ to $u$ fails. This is equivalent to throwing $(D_{max} - 1)kl^2$ balls into $lp$ bins. The maximum number of bins, in which there are at least $m$ balls, is $\left[\frac{(D_{max} - 1)kl^2}{m}\right]$. Thus, the second term in the right hand side of (6) represents the maximum number of time slots of $v$, in which there are more than $m - 1$ neighbors of $u$ other than $v$ itself also transmitting. The first term in the right hand side of (6) represents the maximum number of time slots of $v$, in which its destination $u$ also transmits. We also note $N_f$ cannot be larger than $lp$, i.e., the number of assigned slots. Taking $D_{max} = 10$, $k = 1$, $l = 2$, and $m = 5$ as an example, we know that an arbitrary node $v$ has at most $kl^2 = 4$ coincidences with one of the other interfering neighbors of its destination $u$. Thus, Node $u$ has at most $(D_{max} - 1)kl^2 = 36$ coincidences with the other $D_{max} - 1$ interfering neighbors of Node $u$. Note that if and only if more than $m - 1 = 4$ out of these 36 coincidences are located in one transmission slot of Node $v$, the transmission of Node $v$ in this slot fails. Thus, 36 coincidences can lead to up to $\left[\frac{(D_{max} - 1)kl^2}{m}\right] = \left[\frac{(10 - 1)2^2}{5}\right] = 7$ such slots. $\blacksquare$

In order to investigate how tight the bound in Theorem 3 is, we run simulations as follows. Given that $k = 1$, $p = 23$, $D_{max} = 10$, $m$ is set to two and four, and $l$ varies from one to ten, in each simulation, we randomly select a polynomial with degree $k$ mod $p$ for Node $v$ and then $D_{max}$ other different such polynomials for Node $u$ and its $D_{max} - 1$ other interfering neighbors from the remaining $p^2 - 1$ polynomials, and calculate the value of $N_f$. For each value of $l$, we simulate 100 runs. As shown in Fig. 4, the bound of $N_f$ is relatively loose, especially when the values of $m$ and $l$ are large. This is what we prefer to see, since it implies that the average number of packet failure is typically less than the bound we get.

$^2$A packet failure results from one or more coincidences. However, a coincidence does not necessarily leads to a packet failure.
III. PROPOSED ALGORITHM

Consider a TDMA network with \( N \) nodes with the \( n \)-MPR capability and the maximum node degree is \( D_{\text{max}} \). Each node transmits according to its TSLV, which is determined by its assigned TSAF group consisting of \( l \) unique polynomials with degree \( k \) mod \( p \) (TSAFs).

To avoid coincidences between any two out of \( l \) polynomials distributed to an arbitrary node, the assignment can be made as follows. Note that there are a total of \( p^{k+1} \) polynomials. It has been proved that the maximum degree of the polynomials equals one (\( k = 1 \)) for most cases [18]. Therefore, without loss of generality, we assume that \( k = 1 \) in the subsequent discussion for simplicity. For the case of \( k = 1 \), there are \( p^2 \) polynomials \( f(x) = ax + b \), where \( a = 0, 1, \ldots, p-1 \) and \( b = 0, 1, \ldots, p-1 \). We divide these polynomials into \( p \) groups according to the value of \( a \), each of which consists of \( p \) polynomials. Each node \( i \) selects a group of \( l \) polynomials as its assigned TSAFs for its exclusive use only. Node \( i \) selects \( f_{i,j}(x) = \left( \left\lfloor \frac{i}{p} \right\rfloor - 1 \right) x + b_{i,j} \), where \( j = 1, 2, \ldots, l \) and \( b_{i,j} = \left( \text{mod} \left( i, \left\lfloor \frac{x}{p} \right\rfloor + 1 \right) \right) l + j - 1 \). Note that the parameters \( p \) and \( l \) are known to each node. Thus, each node selects its \( l \) polynomials in a distributed manner according to its identification number. Taking \( N = 3 \), \( l = 2 \), and \( p = 3 \) as an example, the assigned polynomials to Node \( i \) are \( f_{i,1}(x) = (i - 1)x \) and \( f_{i,2}(x) = (i - 1)x + 1 \). As the number of nodes in the network \( N \) grows, more codes are required. Thus, a larger value of parameter \( p \) is chosen to satisfy that the codes assigned to one node are orthogonal to each other. That is, the design parameter \( p \) is chosen such that each node can find its \( l \) polynomials, which will be discussed later.

Since any two polynomials of an arbitrary node has no coincidence, each node can transmit in \( lp \) time slots per frame. Thus, the throughput \( G \) is:

\[
G = \frac{lp - N_f}{p^2},
\]

and the minimum throughput \( G_{\text{min}} \) can be obtained by Theorem 3,

\[
G_{\text{min}} = \frac{lp - N_{f}^{\max}}{p^2}.
\]  

In order to guarantee that our algorithm works correctly, the following two constraints must be satisfied:

\[
\left\lfloor \frac{p}{l} \right\rfloor \geq N_l, \quad lp > N_{f}^{\max}.
\]

Equation (7) must be satisfied to ensure that any two out of \( l \) unique polynomials assigned to an arbitrary node has no coincidence. That is, each node can have \( lp \) assigned time slots for transmissions. Equation (8) states that the total number of transmissions of an arbitrary node is larger than the maximum number of possible packet failures in one frame. Note that (7), (8) ensure that each node can be assigned \( l \) orthogonal codes.

Theorem 4 discusses how to obtain the maximum value of \( G_{\text{min}} \).

**Theorem 4:** Let \( p_1 \) be the largest prime smaller than or equal to \( \frac{2N_{f}^{\max}}{l} \) and \( p_2 \) be the smallest prime larger than \( \frac{2N_{f}^{\max}}{l} \), respectively. The optimal value of \( p \) that achieves the maximum value of \( G_{\text{min}} \) is \( p = \arg\max\{G_{\text{min}}\} \), where \( \arg\max\{G_{\text{min}}\} \) satisfies (7), or is the smallest prime satisfying (7) otherwise.

**Proof:** In order to find the maximum value of \( G_{\text{min}} \), we have to solve (8):

\[
\frac{\partial G_{\text{min}}}{\partial p} = 0.
\]

By (6), (8) can be rewritten as

\[
-lp^{-2} + 2N_{f}^{\max} p^{-3} = 0.
\]

\[
-3N_{f}^{\max} p^{-3} \leq 0.
\]

Thus,

\[
p = \frac{2N_{f}^{\max}}{l}.
\]

Since

\[
\frac{\partial^2 G_{\text{min}}}{\partial p^2} \bigg|_{p=\frac{2N_{f}^{\max}}{l}} = -\frac{l^4}{8(l_{\text{max}} - 1)^3} < 0, \quad G_{\text{min}} \text{ increases with } p \text{ when } p < \frac{2N_{f}^{\max}}{l}, \quad \text{and decreases with } p \text{ when } p \geq \frac{2N_{f}^{\max}}{l}.
\]

Since \( p \) is a prime such that (7) and (8) have to be satisfied, we can thus prove Theorem 4.

When \( m = 1 \), it degenerates to the collision-based reception model. Our result thus reduces to the one in [18].

Considering the case that the maximum value of \( G_{\text{min}} \) is achieved when \( p = \frac{2N_{f}^{\max}}{l} \), we have:

\[
\max(G_{\text{min}}) = \frac{l^2}{4 \left(k_{l}^2 + \left\lfloor \frac{(D_{\text{max}} - 1)k l^2}{m} \right\rfloor \right)}.
\]

Thus, we can conclude that the maximum value of \( G_{\text{min}} \) generally remains the same with varying \( l \). However, the average throughput of our algorithm highly depends on the value of \( l \), i.e., the number of assigned polynomials distributed to each node, that is discussed as follows.
A. Average Throughput

We study the average throughput, defined as the average number of successful transmissions per node per slot, of our algorithm. Given that each node is assigned \( l \) TSAFs (polynomials) with the \( m \)-MPR capability, we give some definitions as follows. Let \( X_i \) and \( X_i^j \) be the numbers of successful transmissions of an arbitrary node determined by its \( i \)-th TSAF in each frame and in Subframe \( j \), respectively. That is, \( i = 1, \ldots, l \) and \( j = 0,1, \ldots, p-1 \). Note that the frame length is \( p^2 \). The average throughput \((G_a)\) can be expressed as follows:

\[
G_a = \frac{E \left[ \sum_{i=1}^{l} X_i \right]}{p^2} = \frac{E \left[ \sum_{i=1}^{l} \sum_{j=0}^{p-1} X_i^j \right]}{p^2} = \frac{l}{p} E \left[ X_i^j \right], \tag{14}
\]

where

\[
E \left[ X_i^j \right] = P \left( X_i^j = 1 \right). \tag{15}
\]

Consider a transmission from Node \( u \) to Node \( v \). Let \( f_u^i(x) \) be the \( i \)-th TSAF assigned to Node \( u \). For \( i = 1, \ldots, l \) and \( j = 0,1, \ldots, p-1 \), we obtain \( E[X_i^j] \) in the following. The event that \( X_i^j = 1 \) happens if and only if both \( A \) and \( B \) happen. \( A \) is the event that Node \( v \) does not transmit in Slot \( f_u^i(j) \) in Subframe \( j \). \( B \) is the event that there are less than \( m \) neighbors of Node \( v \), other than Node \( u \) itself, transmitting in Slot \( f_u^i(j) \) in Subframe \( j \). Suppose that Events \( A \) and \( B \) are independent (referred to as Assumption 1). We have:

\[
P \left( X_i^j = 1 \right) = P \left( A \right) P \left( B \right). \tag{16}
\]

Node \( v \) transmits in \( l \) different time slots in Subframe \( j \). We assume that these \( l \) slots are randomly selected from Slot 0 to Slot \( p-1 \) in Subframe \( j \) (referred to as Assumption 2). Thus, the probability that \( A \) happens is:

\[
P \left( A \right) = 1 - \frac{\binom{p-1}{l-1}}{\binom{p}{l}}. \tag{17}
\]

Let \( N^r \) denote the number of ways, given a TSAF of Node \( u \), \( f_u^i(x) \), for selecting \( l(D_{\text{max}}-1) \) other TSAFs, \( r \) out of which has the same value \( f_u^i(j) \) in Subframe \( j \). There are \( \binom{p^2-1}{l(D_{\text{max}}-1)} \) ways to select \( l(D_{\text{max}}-1) \) TSAFs from the remaining \( p^2-1 \) TSAFs. Thus, the probability that \( B \) happens is as follows:

\[
P \left( B \right) = \frac{\sum_{r=0}^{m-1} N^r}{\binom{p^2-1}{l(D_{\text{max}}-1)}}. \tag{18}
\]

Given \( j \), we can categorize the \( p^2 \) TSAFs into \( p \) groups according to their values in Subframe \( j \). That is, Group \( G_n \), where \( n = 0,1, \ldots, p-1 \), contains those TSAFs, the values of which in Subframe \( j \) are \( n \). Note that TSAFs over GF\((p)\) are uniformly distributed over \( \{0,1,2, \ldots, p-1\} \). Thus, \( |G_n| = p \), where \( n = 0,1, \ldots, p-1 \). Thus, the value of \( N^r \) (\( r = 0,1, \ldots, m-1 \)) is given as follows:

\[
N^r = \binom{p-1}{r} \frac{p^2-p}{l(D_{\text{max}}-1)-r}. \tag{19}
\]

In order to validate our analytical results, we run simulations without making Assumptions 1 and 2. Each simulation data value is obtained by averaging the results of 100 simulation runs, and the 95% confidence intervals are shown on each of the simulated point. The simulation results demonstrate that the two assumptions do not affect the accuracy of our analytical results. The simulations were performed on a regular graph model [29], in which we put all \( N \) nodes around a circle, and connect each to its \( D_{\text{max}} \) nearest neighbors on either side as neighbors. Each node randomly selects one of its neighbors as the destination of its packets. Given that \( N = 100 \) and with the 4-MPR capability, we can observe that the simulations match our analytical results well. In Fig. 5, we can also observe that the average throughput increases with increasing \( l \) when \( D_{\text{max}} \) is small, but remains almost the same with \( l \) greater than a certain value when \( D_{\text{max}} \) is large. This implies that there exists a threshold value of \( l \), beyond which, the increase in the number of collisions is almost the same as that in the number of transmission slots introduced by assigning more TSAFs to a node. The threshold is larger when \( D_{\text{max}} \) is smaller, and vice versa. Thus, the value of \( l \) can be chosen according to this principle.

B. Algorithm Description

Based on the aforementioned discussions, we can propose the \( m \)-MPR-\( l \)-code topology-transparent scheduling algorithm with the \( m \)-MPR capability as below:

1) Use Theorem 4 to select the value of \( p \) for the given \( N \), \( D_{\text{max}} \), and \( m \) such that the minimum guaranteed throughput is maximized.

\footnote{If \( D_{\text{max}} \) is odd, nodes which are opposite with each other are considered as neighbors.}
2) Based on the average throughput expression in (20), choose the proper value of \( l \) that maximizes the average throughput (20) by exhaustive search for the given values of \( N \), \( D_{\text{max}} \), and \( m \).

3) Each node is randomly assigned \( l \) different TSAFs, any two of which has no coincidence.

4) Each node calculates its TSLV, which consists of \( l p \) time slots for possible transmissions, according to its assigned TSAFs.

5) Each node transmits its data packets at its assigned slots.

Some discussions of the proposed algorithm are as follows. Firstly, the proposed algorithm is rather simple. Steps 1 and 2 of the proposed algorithm can be done offline at the network initialization phase. The most time-consuming step is Step 2, since it requires exhaustive search of \( l \) to achieve the maximum average value of the throughput. However, as discussed at the end of Section III-A, there exists a threshold value of \( l \), beyond which, the increase in the number of collisions is almost the same as that in the number of transmission slots introduced by assigning more TSAFs to a node. The threshold is larger when \( D_{\text{max}} \) is smaller, and vice versa. Thus, the number of rounds of search of \( l \) is actually limited. We can observe in Fig. 5 that we only need to search six values of \( l \) (from one to six) to achieve the maximum average throughput when \( D_{\text{max}} = 20 \). Thus, its complexity is rather small. Secondly, the proposed algorithm is distributed when the network operates, if no new nodes will join the network after the network initialization phase. Before the network operates, a central entity runs Steps 1 and 2 of the algorithm and tells the values of \( p \) and \( l \) to all nodes. After that, the central entity is not necessary anymore. Then, each node determines its transmission time slots in a fully distributed manner according to Steps 3 and 4 of the algorithm. Each node transmits and receives packets according to Step 5 and the transmission time slots of each node will not change when the network operates. Our algorithm works correctly when the network topology changes. The reasons are as follows. \( D_{\text{max}} \) is an estimated parameter. As mentioned in Section II, \( D_{\text{max}} \) is always pessimistically estimated according to empirical data to ensure that the actual number of interfering neighbors does not exceed the estimate. We also show that the degradation of the performance of our algorithm is rather small when the estimation of \( D_{\text{max}} \) is not so accurate in Section IV.

Moreover, although sometimes some nodes may disappear, leave, and re-join the network, we assume that the total number of nodes in the network, \( N \), is a known parameter and remains constant. This is a general and practical assumption in most previous work focusing on wireless networks [5], [7], [9], [13], [15]–[17], [23], [34]. Note that nodes may join the network after the network initialization phase in some cases. In such cases, a central entity is required in our algorithm to maintain which codes are used by the existing nodes and which codes can be assigned to the new nodes. When a new node joins the network, it communicates with the central entity and gets its codes [1]. Thus, our algorithm cannot be considered as a fully distributed scheduling algorithm in this case. However, our algorithm cannot be simply considered as a centralized algorithm, since there is a significant difference between our algorithm and the traditional centralized scheduling algorithm [2], [9], [13], [23], [27]. In the traditional centralized scheduling algorithms, the central entity has to know the detailed network connectivity and traffic information other than the information of each node. When the central entity fails, the algorithms cannot work correctly in mobile networks. In contrast, the central entity in our algorithm only needs to maintain which codes are used by the existing nodes and which codes can be assigned to the new nodes, which is a much easier task. When the central entity fails, the existing nodes can still transmit correctly, although the new nodes may not join the network correctly. Especially in the case that no node will join the network after the network initialization phase, the central entity is not necessary in our algorithm after the initialization phase. We also show that even if the value of \( N \) changes as the network operates, the degradation of the performance of our algorithm is very small in Section IV.

C. Delay Analysis

In this subsection, we use a discrete time \( M/G/1 \) queuing model in which the first customer of each busy period receives exceptional service [30] to approximate the average packet delay in our algorithm. The simulation results in Section IV show that the approximation does not affect the accuracy of our analysis. Packets arrive at an infinite buffer in a Poisson fashion with rate \( \lambda \) packets per slot. We assume that arrivals occur just after the beginning of a slot, departures take place just before the end of a slot.

Considering a given node as a server, its service slots are fixed. That is, the distribution of the service time of the packets arriving when the server is busy (there are other packets queued or being served) is different from that of the packets arriving when the server is idle (there are no packets queued or being served). We assume that the service time distribution of a packet which initiates a busy period is \( B_1(x) \) with first and second moments \( b_1 \) and \( b_2 \). The distribution of service time of all subsequent packets in the same busy period is \( B_2(x) \) with first and second moments \( b_3 \) and \( b_4 \), which is independent of \( B_1(x) \). Note that the service time of the packet that initiates a busy period is less than that of the subsequent packets on average. That is, \( b_1 < b_2 \). Thus, the average delay can be expressed as follows:

\[
E[W] = E[W|\text{Idle}]P_0 + E[W|\text{Busy}](1 - P_0),
\]

where \( P_0 \) is the probability that the system is empty. According to the results obtained in [30], we have:

\[
P_0 = \frac{1 - \lambda b_2}{1 - \lambda(b_2 - b_1)}. \tag{22}
\]

According to the Pollaczek–Khinchine Formula, the value of \( E[W|\text{Busy}] \) can be obtained as follows:

\[
E[W|\text{Busy}] = \frac{\lambda b_2 \left( 1 + \frac{b_1}{b_3} \right)}{2 \left( \frac{1}{b_2} - \lambda \right)} + b_2. \tag{23}
\]

In the following, we calculate \( \tilde{b}_2 = \frac{1}{\mu} \). \( \mu \) is the probability that the packet can be transmitted successfully in a slot.
Consider a transmission from an arbitrary node \( u \) to its neighbor \( v \). Given an arbitrary slot \( t \), let \( C \) be the event that \( t \in TSLV_u \), \( D \) the event that \( t \in TSLV_v \) and Node \( v \) has packets to transmit, and \( E \) the event that less than \( m \) neighbors of Node \( v \) other than \( u \) itself have packets to transmit in Slot \( t \). Thus, we have:

\[
\mu = P(C)P(D)P(E).
\]  

According to the calculation in Section III-A, we have:

\[
P(C) = \frac{l}{p^l},
\]

and

\[
P(D) = \left[ 1 - \left( \frac{l-1}{l} \right) \right]^m + \left( \frac{l-1}{l} \right)^m (1 - \rho),
\]

where \( \rho \) is the probability that the node has packets to transmit and \( \rho = 1 - P_0 \).

Moreover, \( P(E) \) can be expressed as follows:

\[
P(E) = \sum_{r=0}^{m-1} \sum_{r=0}^{D_{max} - 1} N^r \sum_{i=0}^{m-1} \left( \frac{l-1}{l} \right)^r \rho^i (1 - \rho)^{r-i}.
\]

Substituting (25)–(27) into (24), we can calculate the value of \( \mu \) numerically, although it is difficult to obtain its closed-form expression. Substituting the value of \( \mu \) into (23), we can obtain the value of \( E[W|Busy] \).

Note that \( E[W|Idle] = \bar{b}_1 \). We can obtain the average packet delay as follows:

\[
E[W] = \bar{b}_1 P_0 + E[W|Busy](1 - P_0).
\]

It is difficult, if not impossible, to calculate the values of \( \bar{b}_2 \) and \( \bar{b}_1 \). However, the detailed calculation of queuing system parameters is not the focus of this work. Thus, we get the values of \( \bar{b}_2 \) and \( \bar{b}_1 \) via simulations. Substituting (22)–(27) into (28), we can get the average packet delay.

### IV. Performance Evaluation

In this section, we quantitatively compare our \( m \)-MPR-l-code topology-transparent scheduling algorithm with the conventional topology-transparent algorithm designed for the collision-based reception model in [18] (known as Ju’s algorithm). We do not include the algorithm in [33] as a comparison, since it is only designed for static networks with star topology. It assumes that a central node knows all the information of other nodes and cannot be applied in mobile ad hoc networks. All simulations have been conducted using Matlab.

We compare our algorithm with the centralized conflict graph coloring algorithm in [27]. In this conflict graph coloring algorithm, a central scheduler knows the detailed network connectivity and traffic information, constructs a conflict graph, and applies the vertex coloring algorithm to compute and distribute the transmission schedule to each node. When the network connectivity and traffic information change, we assume that the central scheduler can get the updated network information, re-compute, and distribute the updated schedules immediately without introducing any overhead. Thus, this ideal algorithm is impractical in mobile networks. However, our algorithm is independent of topology changes, which can be easily implemented in mobile networks. We also compare our algorithm with slotted ALOHA with the \( m \)-MPR capability [16], [17], although the latter algorithm cannot offer any throughput guarantees.

#### A. Simulation Setup

We conduct simulations on two graph models, namely, the geometric model for the average performance and the regular graph model [29] for the worst performance. In the geometric model, we adopt the Gauss-Markov mobility model, which has been shown to be more realistic than the widely used Random Waypoint model [3]. All nodes are uniformly and randomly distributed in a region of 1000 m \( \times \) 1000 m initially. The tuning parameter \( \theta \) is used to present different levels of randomness in the Gauss-Markov model. We set \( \theta = 0.5 \). Given \( D_{max} \), we set the interference range of each node \( R_I \) such that the probability that the number of interfering neighbors of an arbitrary node exceeding \( D_{max} \), which is

\[
\sum_{i=D_{max}+1}^{N-1} \binom{N-1}{i} \left( \frac{\pi R_I^2}{A} \right)^i \left( 1 - \frac{\pi R_I^2}{A} \right)^{N-1-i},
\]

is smaller than 0.05. For example, \( R_I = 140 \) m if \( (N, D_{max}) = (100, 10) \). In the regular graph model, the degree of each of the \( N \) nodes is set to \( D_{max} \). In other words, each node has exactly \( D_{max} \) interfering neighbors. Thus, the average number of interfering neighbors in the geometric model is less than that in the regular graph model with the same \( D_{max} \).

We apply the optimal frame structure derived in Theorem 4. \( l \) is set to different values for different network parameters, as discussed later in details. For each simulation point, we conduct a simulation run for 100 continuous frames in the geometric model and 100 randomly generated topologies in the regular graph model, unless stated otherwise. The 95% confidence intervals are also drawn in the figures.

#### B. Simulation Results

1) Effect of \( D_{max} \) on Average Throughput: Given that \( N = 100 \), we investigate the performance of our \((m, l)\)-TTS algorithm with different \( D_{max} \) settings from 6 to 40. A larger \( D_{max} \) indicates that the network is denser and there are more possible conflicts. In the regular graph model, we set \( l = m + 2 \). However, we set \( l = m + 6 \) in the geometric model, since the average number of interfering neighbors under the geometric model is smaller than that under the regular model. We can see that the average throughput of our algorithm is about 60% of that of the ideal coloring algorithm. However, the ideal coloring algorithm is impractical, since it is a centralized algorithm. The cost of schedule recomputation is unacceptable in mobile networks, since the network connectivity and traffic information change frequently. As shown in Figs. 6 and 7, we can see that our algorithm outperforms slotted ALOHA for

\[\]

In fact, this ideal algorithm is impractical even in static networks, since the traffic load of each node changes as time evolves, resulting in costly recomputation and distribution of new schedules.

\[\]

The movement of each node follows the Gauss–Markov mobility model.
most cases, especially when the $D_{\text{max}}$ is large. This is different from the case in the collision-based reception model. It has been shown [24] that the average throughput of topology-transparent scheduling algorithm is not as good as that of slotted ALOHA under the collision-based reception model. The superiority of our algorithm in the MPR model implies that our algorithm can take advantage of the powerful MPR capability better. We can observe that the average throughput of the nodes in the geometric model is better than that of the nodes in the regular graph model. This is reasonable, since the average node degree of the geometric model is smaller than that of the regular graph model. Moreover, the superiority of our algorithm over slotted ALOHA is greater in the geometric model than that in the regular graph model. This is because the nodes cannot select the optimal transmission probability in the geometric model, since they do not know the number of its neighbors.

As shown in Figs. 6 and 7, our algorithm dramatically outperforms the conventional topology-transparent scheduling algorithm. We can see that the average throughput of the conventional topology-transparent scheduling algorithm remains almost the same when $m$ increases, implying that it cannot take full advantage of the MPR capability.

2) Effect of Inaccuracies in Estimating $D_{\text{max}}$ and $N$ on Performance: We use the performance ratio as the metric to investigate the effect of inaccuracies in the estimation of $D_{\text{max}}$ on the average throughput of the proposed algorithm. The performance ratio is defined as the average throughput under the actual $D_{\text{max}}$ divided by that under the design value of $D_{\text{max}}$. Fig. 8 shows the simulation results under the regular graph model and the geometric model, respectively.

Given that the design parameters $(N, D_{\text{max}})$ is (100, 20), we study the effect of inaccuracies in the estimation of $D_{\text{max}}$ on the average throughput of the proposed $(m, l)$-TTS algorithm, where $m$ is set to two and four. According to Theorem 4, the optimal values of $p$ for different parameters are shown in Table II. By varying the actual maximum node degree from 6 to 40, we can observe in Fig. 8(a) that the performance ratio is smaller (larger) than one when the actual maximum node degree is smaller (larger) than that of the design value.
degree is larger (smaller) than the design value of $D_{\text{max}}$. The performance ratio decreases with increasing actual maximum node degree. The performance ratio is close to one when the actual maximum node degree is close to the design value ($D_{\text{max}} = 20$). As the difference between the actual and design values of $D_{\text{max}}$ increases, the penalty on the network performance increases (when the actual value of $D_{\text{max}}$ is larger than its design value). We can also observe that the performance ratio of (4, 6)-TTS algorithm is more sensitive to the difference between the actual and design values of $D_{\text{max}}$ than that of (2, 4)-TTS algorithm under the regular graph model. This is due to the fact that the maximum number of packet failures of (4, 6)-TTS algorithm increases (decreases) faster with the increasing (decreasing) actual value of $D_{\text{max}}$ than that of (2, 4)-TTS algorithm. The reason is explained as follows. The increased maximum number of packet failures in one frame is $\frac{4 \times 4 \times \Delta_D}{2} = 8 \Delta_D$ for $(m, l) = (2, 4)$, but $\frac{6 \times 6 \times \Delta_D}{4} = 9 \Delta_D$ for $(m, l) = (4, 6)$, where $\Delta_D$ is the difference between the actual and design values of $D_{\text{max}}$. Similar phenomena are also exhibited in Fig. 8(b). Comparing Fig. 8(a) and (b), we can see that the effect of inaccuracies in the estimation of $D_{\text{max}}$ on the performance in the geometric model is much smaller than that in the regular graph model. The average node degree is typically smaller than the maximum node degree $D_{\text{max}}$ in reality. Thus, we conclude that our algorithm is insensitive to the accuracy in the estimation of $D_{\text{max}}$.

Fig. 8 studies the effect of inaccurate estimation of $N$ on the average throughput. In Fig. 9, we use $(N, D_{\text{max}}) = (100, 20)$ as our design values. The corresponding optimal values of $p$ are shown in Table II. By varying the actual number of nodes $N$ from 100 to 800, we can observe in Fig. 9 that the performance ratio equals one for most cases, indicating that an inaccurate estimation of $N$ has almost no effect on the performance. The reason is as follows. According to the discussion in Section III and (9), our algorithm can support up to $\left\lceil \frac{p}{4} \right\rceil$ nodes. Thus, the corresponding maximum number of nodes in the network that our algorithm can support is listed in Table III. The only exception happens when the actual number of nodes, in the (4, 6)-TTS algorithm, is 800, which is larger than the maximum number of nodes supported. In this case, some nodes may be assigned the same polynomials. If these nodes with the same polynomials are within the interference range of each other, their transmissions fail. However, the situation that the actual value of $N$ (800) is much larger than the designed value (100) may never happen in well-designed networks. Thus, we conclude that the effect of inaccurate estimation of $N$ on the performance is very small and can be ignored.

3) Supporting Heterogeneous Traffic: Given $N$, we investigate the performance of our $(m, l)$-TTS algorithm with different settings on $D_{\text{max}}$ in heterogeneous networks. Different number of codes can be assigned to provide different QoS for different classes of nodes in heterogeneous networks. Without loss of generality, we suppose $N$ nodes are divided into two classes, namely, Class $V_1$ with a higher QoS requirement and Class $V_2$ with a lower QoS requirement. The number of nodes in Class $V_1$ and $V_2$ is $\lfloor \beta N \rfloor$ and $N - \lfloor \beta N \rfloor$, respectively, where $0 < \beta < 1$. Here, we assume that $\beta = 0.5$. $l_1$ and $l_2$ codes are assigned to the nodes in Classes $V_1$ and $V_2$, respectively, where $l_1 = 2l_2$. Note that the maximum value of packet failures for a transmission from an arbitrary node in Classes $V_1$ and $V_2$ is $N_{f, V_1}^{\text{max}} = kl_1^2 + \left[ \frac{(D_{\text{max}} - 1)kl_1^2}{m} \right]$ and $N_{f, V_2}^{\text{max}} = kl_1l_2 + \left[ \frac{(D_{\text{max}} - 1)kl_1l_2}{m} \right]$, respectively. Thus, the average minimum guaranteed throughput is as follows:

$$G_{\text{min}}^h = \beta \frac{l_1p - N_{f,V_1}^{\text{max}}}{p^2} + (1 - \beta) \frac{l_2p - N_{f,V_2}^{\text{max}}}{p^2},$$

where $p = \frac{\beta}{\beta - 1}$.

![Fig. 8](image-url) The effect of inaccuracies in the estimation of $D_{\text{max}}$ on the throughput. (a) Regular graph model. (b) Geometric model.

### Table II

<table>
<thead>
<tr>
<th>m</th>
<th>Regular graph model</th>
<th>Geometric model</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>83</td>
<td>167</td>
</tr>
<tr>
<td>4</td>
<td>67</td>
<td>113</td>
</tr>
</tbody>
</table>
where the following constraints must be satisfied:

\[
\begin{align*}
\left\lfloor \frac{p}{l_1 + l_2} \right\rfloor p & \geq \frac{N}{2}, \\
l_1 p & > N_{f,V_1}^{\text{max}}, \\
l_2 p & > N_{f,V_2}^{\text{max}}.
\end{align*}
\]

Let \(p_3\) be the largest prime smaller than or equal to \(\frac{N_{f,V_1}^{\text{max}} + N_{f,V_2}^{\text{max}}}{l_2}\) and \(p_4\) be the smallest prime larger than \(\frac{1}{l_2}\), respectively. Thus, following the proof of Theorem 4, we obtain the optimal value of \(p\) that achieves the maximum value of \(G_{h_{\text{min}}}\) is \(p = \arg \max(G_{\text{min}})\), where \((p_3, p_4)\) satisfying (26)–(28) otherwise.\(^6\)

We set \(N = 100\), \(\beta = 0.5\), and vary \(D_{\text{max}}\) from 6 to 40, \(l_2 = m + 2\), \(l_1 = 2l_2\), where \(m\) is set to two and four. We study the average throughput of the nodes in Classes \(V_1\) and \(V_2\) in the geometric model. As shown in Fig. 10, our algorithm works well as well supporting heterogeneous traffic. The simulation results for the regular graph model follow the same trend, and are thus omitted due to space limitations.

4) Effect of \(m\)-MPR Capability on Minimum Guaranteed Throughput and Average Throughput: We study the effect of the \(m\)-MPR capability on the minimum guaranteed throughput and average throughput under two different \(m\)-MPR model, namely, the strong \(m\)-MPR model in (3), (4) and the \(m\)-MPR model in [22] achieved by the blind interference cancellation algorithm using polynomial phase-modulating sequences (referred to as PPS \(m\)-MPR model). In the PPS \(m\)-MPR model, \(C_{n,i}\) is given as follows:

\[
C_{n,i} = \frac{F_{n,i}(C)}{C^n},
\]

if \(n = 1, 2, \ldots, m\) and \(i = 0, 1, \ldots, n\), and \(C_{n,i} = 0\) otherwise. \(C\) is the size of a codebook (the number of polynomial phase-modulating sequences) and \(F_{n,i}(C)\) is the number of ways that exactly \(i\) out of \(n\) taken codes are different. Interested readers are referred to (29)–(34) in [22] for the detailed expression of \(C_{n,i}\). We set \(C = 32\) according to the discussion in [22].

Fig. 9. The effect of inaccuracies in the estimation of \(N\) on the throughput. (a) Regular graph model. (b) Geometric model.

Fig. 10. Performance of our algorithm supporting heterogeneous traffic.

---

\textbf{TABLE III}

\begin{tabular}{|c|c|c|}
\hline
\textbf{m} & \textbf{Regular graph model} & \textbf{Geometric model} \\
\hline
\hline
2 & 1660 & 3340 \\
4 & 737 & 1243 \\
\hline
\end{tabular}

---

\(^6\)Our algorithm can also support more than two classes of nodes in heterogeneous networks by choosing the proper value of \(p\) according to the aforementioned discussion.
5) Packet Delay and End-to-End Performance: Given that \( N = 100 \) and \( D_{\text{max}} = 20 \), we investigate the average packet delay of our \((m, l)\)-TTS algorithm with different settings of the values of \( \lambda \) and \( m \). In order to validate our analytical results, we run simulations under the regular graph model [29], in which each node has \( D_{\text{max}} \) interfering neighbors. We run each simulation for 50 data frames. In the simulation, we assume that each queue has infinite capacity and uses First-In-First-Out (FIFO) service discipline. The destination of a packet is randomly chosen from its neighbors. The transmission slots of each node are assigned deterministically using our \((m, l)\)-TTS algorithm. We set \( l = m + 2 \). According to Theorem 4, the values of \( p \) in the \((2, 4)\)-TTS algorithm and \((4, 6)\)-TTS algorithm are 83 and 67, respectively. As shown in Fig. 13, our analytical results match well with those from our simulations.

We investigate the end-to-end performance of our algorithm in terms of throughput. The average end-to-end throughput of scheduling algorithms highly depends on the network topologies and routing protocol applied. The design and implementation of the topology control and routing protocols that can balance the network loads to optimize the network performance are out of the scope of this paper. Thus, we use a simple regular graph model in the simulations, in which each node connects the nearest \( D_{\text{max}} \) nodes as its neighbors. Let \( S \) and \( \text{Dest} \) be the sets of source nodes and destination nodes, respectively. \( 2N_{\text{SD}} \) out of \( N \) nodes are randomly selected. The first \( N_{\text{SD}} \) nodes belong to \( S \), and the other \( N_{\text{SD}} \) ones belong to \( \text{Dest} \). A pair of a source \( S(i) \) and a destination \( \text{Dest}(i) \) is called a source-destination (S-D) pair, where \( i = 1, 2, \ldots, N_{\text{SD}} \). The Bellman-Ford algorithm [10] is used as the routing protocol to calculate the shortest path between each S-D pair. For each of \( N_{\text{SD}} \) source nodes, the arrivals follow a Poisson distribution with rate \( \lambda \) packets per slot. We define the average end-to-end throughput as the average number of packets delivered from sources to their destinations per slot. We set \( (N, D_{\text{max}} = (100, 20)), (m, l) = (4, 6) \), and \( N_{\text{SD}} \) to be 10, 20, and 50. Varying \( \lambda \) from 0.01 to 0.1, we run each simulations for 50 data frames. As shown in Fig. 14, the average end-to-end throughput increases almost linearly with the arrival rate when \( \lambda \) is small. When the arrival rate becomes larger, the end-to-end throughput increases more slowly or remains the same. Comparing Figs. 7(b) and 14, we can see that when the number of S-D pairs is small, the end-to-end throughput is comparable to the one-hop throughput shown in Fig. 7(b). This is due to the fact that only a small fraction of nodes have packets to transmit when the number of S-D pairs is small. Thus, the number of interfering neighbors of each node here is much smaller than that in the simulations of Fig. 7(b). With the increasing number of S-D pairs, the average end-to-end throughput decreases, especially when the arrival rate is large.

V. Conclusion

In this paper, we propose a novel \( m \)-MPR-\( l \)-code topology-transparent scheduling ((\( m, l \))-TTS) algorithm for mobile ad hoc networks with the MPR capability. As far as we know, this is the first schedule-based algorithm with MPR, which is independent of topology changes. Our algorithm is easy to implement and is oblivious to the network topology changes, because it only needs global parameters such as the number of nodes and the maximum node degree in the network. Moreover, our algorithm provides the minimum guaranteed throughout to each participating node. The proposed algorithm greatly improves the minimum guaranteed throughput and average throughput of our algorithm by assigning multiple polynomials to each node to take full advantage of the powerful MPR capability. The performance improvement of our \((m, l)\)-TTS algorithm over the conventional topology-transparent scheduling algorithms designed for the collision-based reception model is
almost linear with increasing $m$. The simulation results show that our algorithm performs better than slotted ALOHA with MPR as well.

REFERENCES


Yiming Liu (S’14–M’14) received the B.Eng. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2009 and 2014, respectively.

He is an Engineer with the China Academy of Electronics and Information Technology, Beijing, China. His research interest is in the area of wireless communications and networking, specifically in transmission scheduling and cross-layer design for wireless multihop networks.

Victor O. K. Li (S’80–M’81–F’92) received the S.B., S.M., E.E., and Sc.D. degrees from MIT, Cambridge, MA, USA, in 1977, 1979, 1980, and 1981, respectively, all in electrical engineering and computer science. He is a Chair Professor of information engineering and the Head of the Department of Electrical and Electronic Engineering at The University of Hong Kong (HKU), Pokfulam, Hong Kong, where he has also served as the Associate Dean of Engineering and the Managing Director of Versitech Ltd., which is the technology transfer and commercial arm of HKU. He served on the board of China.com Ltd., and now, he serves on the board of Supervision Holdings Ltd. and Anxin-China Holdings Ltd., which are listed on the Hong Kong Stock Exchange. Previously, he was a Professor of electrical engineering at the University of Southern California (USC), Los Angeles, CA, USA, where he was also the Director of the USC Communication Sciences Institute. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world. He has received numerous awards, including the PRC Ministry of Education Changjiang Chair Professorship at Tsinghua University, the U.K. Royal Academy of Engineering Senior Fellowship in Communication, the Croucher Foundation Senior Research Fellowship, and the Order of the Bronze Bauhinia Star, Government of the Hong Kong Special Administrative Region, China. He is a Registered Professional Engineer and Fellow of the Hong Kong Academy of Engineering Sciences, the IAE, and the HKIE.

Ka-Cheong Leung (S’95–M’01) received the B.Eng. degree in computer science from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 1994 and the M.Sc. degree in electrical engineering (computer networks) and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, CA, USA, in 1997 and 2000, respectively. From 2001 to 2002, he was a Senior Research Engineer at Nokia Research Center, Nokia Inc., Irving, TX, USA. Between 2002 and 2005, he was an Assistant Professor at the Department of Computer Science, Texas Tech University, Lubbock, TX. Since June 2005, he has been with The University of Hong Kong, where he is currently an Assistant Professor in the Department of Electrical and Electronic Engineering. His research interests include transport layer protocol design, wireless packet scheduling, congestion control, routing, and vehicle-to-grid (V2G).

Lin Zhang received the B.Sc., M.Sc., and Ph.D. degrees from Tsinghua University, Beijing, China, in 1998, 2001, and 2006, respectively. He is currently a Visiting Associate Professor at Stanford University, Stanford, CA, USA, and an Associate Professor at Tsinghua University, where he has been happily teaching selected topics in communication networks and information theory to senior undergraduate and graduate students. He is a co-author of more than 40 peer-reviewed technical papers and five U.S. or Chinese patent applications. His current research focuses on wireless sensor networks, distributed data processing, and information theory.

In 2006, he led a 2008 Beijing Olympic Stadium (the “Bird’s Nest”) structural security surveillance project, which deployed more than 400 wireless temperature and tension sensors across the stadium’s steel support structure and dome. The system adopted a flexible spectrum sensing and adaptive multihop routing algorithm, to overcome strong radio interference and long-distance transmission channel-fading, and played a critical role in the construction of the stadium. Since then, he has implemented wireless sensor networks in a wide range of application scenarios, including underground mine security, precision agriculture, and industrial monitoring.

Since 2008, he has been working in close association with CISCO to develop a Metropolitan Area Sensing and Operating Network (MASON), which provides a smart-city and intelligent-urbanization sensor network system for metropolitan areas and has attracted the interest of several large-sized Chinese cities, including Beijing, Shenzhen, Tianjing, and Chengdu. Recently, he also has led two National Science Foundation of China projects, three National High-Tech Developing (863) projects, and more than ten research projects funded primarily by private industry in the area of wireless sensor networks.

Dr. Zhang and his team were the recipients of the IEEE/ACM SenSys 2010 Best Demo Awards. In 2004 and 2010, he was a recipient of the Excellent Teacher Awards from Tsinghua University.