High Spatial Reuse Link Scheduling Algorithms for STDMA Wireless Ad Hoc Networks

Ashutosh Deepak Gore  Srikanth Jagabathula  Abhay Karandikar

Department of Electrical Engineering
Indian Institute of Technology - Bombay
Email: adgore@ee.iitb.ac.in

IITM-IITB Communications Workshop, July 2006
1 Introduction

2 System Model

3 SINR-Based Algorithms

4 Performance Results

5 Analytical Results

6 Discussion
Wireless Ad Hoc Network

- Nodes communicate via the broadcast wireless medium
- Communication is successful if SINR at receiver exceeds a certain threshold
- Practical limitations on maximum transmission power
- Contention is location-dependent
- T/F/C-DMA and CSMA/CA suitable only for single-hop communication and infrastructure-based wireless networks
- Throughput of multihop wireless networks can be increased by spatial reuse, i.e., concurrent communication between reasonably distant s-d pairs
- Spatial reuse further necessitated by scarcity and cost of wireless spectrum
Spatial Time Division Multiple Access (STDMA)

- Extension of TDMA wherein multiple s-d pairs can communicate in the same slot
- STDMA schedule
  - sequence of slots s.t. every s-d pair is assigned a unique slot
  - communicating entities assigned to the same slot do not collide
  - schedule repeats periodically during the operation of the network
- Types of STDMA scheduling
  - Broadcast scheduling
  - Link scheduling
- Centralized link scheduling
Assumptions

- STDMA wireless network with a finite number of immobile nodes in a 2-D plane
- A node can transmit, receive or remain idle in a timeslot
- Homogeneous nodes
- Backlogged nodes
- Isotropic transmission and propagation environment
  - omnidirectional antennas
  - fading and shadowing effects not considered
- A node communicates only with a single neighbor using unicast
The physical model of the system is completely described by
\( \Phi(N, (r_1, \ldots, r_N), P, \gamma_c, \gamma_i, \alpha, N_0) \), where

\[
\begin{align*}
N &= \text{number of nodes} \\
r_i &= (x_i, y_i) = \text{Cartesian coordinates of } i^{th} \text{ node} \\
P &= \text{transmission power of every node} \\
\gamma_c &= \text{communication threshold} \\
\gamma_i &= \text{interference threshold} \\
\alpha &= \text{path loss exponent} \\
N_0 &= \text{thermal noise power}
\end{align*}
\]
A link schedule for \( \Phi(\cdot) \) is denoted by \( \Sigma(C, S_1, \cdots, S_C) \), where

\[
\begin{align*}
C & = \text{number of colors (slots) in the link schedule} \\
M_i & = \text{number of concurrent s-d pairs in } i^{th} \text{ slot} \\
S_i & = \text{set of concurrently communicating s-d pairs in } i^{th} \text{ slot} \\
& := \{ t_{i,1} \rightarrow r_{i,1}, \cdots, t_{i,M_i} \rightarrow r_{i,M_i} \} \\
t_{i,j} \rightarrow r_{i,j} & = \text{transmission from node } t_{i,j} \text{ to node } r_{i,j} \text{ in } i^{th} \text{ slot}
\end{align*}
\]
Consider a link schedule $\Sigma(\cdot)$ for the STDMA network $\Phi(\cdot)$. According to the physical interference model, $t_{i,j} \rightarrow r_{i,j}$ is successful if

$$\frac{P}{N_0 + \sum_{k=1}^{M_i} \frac{P}{D^\alpha(t_{i,k},r_{i,j})}} \geq \gamma_c$$

where $D(a,b)$ is the Euclidean distance between nodes $a$ and $b$. Suppose there is only one transmission in $i^{th}$ timeslot; then $t_{i,1} \rightarrow r_{i,1}$ is successful if

$$D(t_{i,1},r_{i,1}) \leq \left(\frac{P}{N_0 \gamma_c}\right)^{\frac{1}{\alpha}} =: R_c$$

where $R_c$ is termed as communication range.
Interference Range

Signal transmitted by $t_{i,j}$ can be received at unintended receiver $r_{i,k}$, $k \neq j$, with received power strong enough to prevent signal from intended transmitter $t_{i,k}$ to be decoded with acceptably low probability of error. Assuming hypothetically that $t_{i,j} \rightarrow r_{i,k}$ is the only transmission in $i^{th}$ timeslot, we impose $\gamma_i < \text{SINR} < \gamma_c$, which translates to

$$R_c < \frac{P}{N_0 \gamma_i} \leq \left( \frac{P}{N_0 \gamma_i} \right)^{\frac{1}{\alpha}} =: R_i$$

where $R_i$ is termed as interference range.
Protocol Interference Model

According to the protocol interference model, $t_{i,j} \rightarrow r_{i,j}$ is successful if:

- the distance $D(t_{i,j}, r_{i,j})$ between these two nodes satisfies
  \[ D(t_{i,j}, r_{i,j}) \leq R_c \]

- any other node $t_{i,k}$ which is within the interference range of the receiver is not transmitting.
  \[ D(t_{i,k}, r_{i,j}) > R_i \quad \forall \; k = 1, \ldots, M_i, \; k \neq j \]
Feasible Schedule

A schedule $\Sigma(\cdot)$ is feasible if it satisfies:

- **Operational constraint:** A node must not perform multiple operations in a single time slot.

  $$\{t_{i,j}, r_{i,j}\} \cap \{t_{i,k}, r_{i,k}\} = \emptyset \ \forall \ 1 \leq j < k \leq M_i \ \forall \ i = 1, \cdots, C$$

- **Communication range constraints:**
  - Every receiver is within the communication range of its intended transmitter.
    $$D(t_{i,j}, r_{i,j}) \leq R_c \ \forall \ j = 1, \ldots, M_i \ \forall \ i = 1, \ldots, C$$
  - Every receiver is outside the communication range of its non-intended transmitters.
    $$D(t_{i,j}, r_{i,k}) > R_c \ \forall \ 1 \leq j < k \leq M_i \ \forall \ i = 1, \cdots, C$$
A schedule \( \Sigma(\cdot) \) is exhaustive if every pair of nodes which are within each other’s communication range in 
\( \Phi(N, (r_1, \ldots, r_N), P, \gamma_c, \gamma_i, \alpha, N_0) \) is included in the schedule twice.

\[ D(a, b) \leq R_c \Rightarrow a \rightarrow b \in \bigcup_{i=1}^{C} S_i \text{ and } b \rightarrow a \in \bigcup_{i=1}^{C} S_i \quad \forall \ 1 \leq a < b \leq N \]
Graph-Based Scheduling Algorithm (GSA)

For purposes of determining schedule $\Sigma(\cdot)$, STDMA network $\Phi(\cdot)$ is modeled by a directed graph $G(\mathcal{V}, \mathcal{E})$, where

$$\begin{align*}
\mathcal{V} &= \{v_1, v_2, \ldots, v_N\} = \text{set of vertices} \\
\mathcal{E} &= \mathcal{E}_c \cup \mathcal{E}_i = \text{set of edges} \\
\mathcal{E}_c &= \text{set of communication edges} \\
\mathcal{E}_i &= \text{set of interference edges}
\end{align*}$$

The mapping from $\Phi$ to $G$ can be described by

$$\begin{align*}
D(a, b) &\leq R_c \quad \Rightarrow \quad v_a \overset{c}{\rightarrow} v_b \in \mathcal{E}_c \quad \text{and} \quad v_b \overset{c}{\rightarrow} v_a \in \mathcal{E}_c \\
R_c &< D(a, b) \leq R_i \quad \Rightarrow \quad v_a \overset{i}{\rightarrow} v_b \in \mathcal{E}_i \quad \text{and} \quad v_b \overset{i}{\rightarrow} v_a \in \mathcal{E}_i
\end{align*}$$
An STDMA link scheduling algorithm is equivalent to assigning a unique color to every communication edge, such that s-d pairs corresponding to communication edges with the same color transmit simultaneously in a particular slot.

A GSA seeks to minimize the total number of colors used to color all edges in $E_c$, subject to:

Any pair of directed communication edges $v_a \rightarrow v_b$, $v_c \rightarrow v_d$ can be colored the same iff:

- there is no primary edge conflict, i.e., vertices $v_a$, $v_b$, $v_c$, $v_d$ are all mutually distinct
- there is no secondary edge conflict, i.e., $v_a \rightarrow v_d \notin E$ and $v_c \rightarrow v_b \notin E$
Limitations of Graph-Based Scheduling Algorithms

- GSAs can be very conservative.
- GSAs can lead to high cumulative interference at a receiver, due to hard-thresholding based on $R_c$ and $R_i$.
- GSAs are not geography-aware.

Exhaustively determining $\Sigma(\cdot)$ which yields the highest throughput under the physical interference model is computationally infeasible.
Motivation

To alleviate these problems, we seek new STDMA link scheduling algorithms which

- yield high throughput under the physical interference model
- have a high degree of spatial reuse
- have low computational complexity
Performance Metrics

Consider the schedule $\Sigma(C, S_1, \ldots, S_C)$ for the physical network $\Phi(N, (r_1, \ldots, r_N), P, \gamma_c, \gamma_i, \alpha, N_0)$. We introduce the following performance metrics:

physical throughput $= \tau = \frac{\sum_{i=1}^{C} \sum_{j=1}^{M_i} I(\text{SINR}_{r_{i,j}} \geq \gamma_c)}{\sum_{i=1}^{C} M_i}$

spatial reuse factor $= \sigma = \frac{\sum_{i=1}^{C} \sum_{j=1}^{M_i} I(\text{SINR}_{r_{i,j}} \geq \gamma_c)}{C}$
To develop low-complexity STDMA link scheduling algorithms with physical throughput close to unity and spatial reuse factor reasonably greater than unity. We only consider STDMA schedules which are feasible and exhaustive.
Abstraction

The core of every STDMA link scheduling algorithm consists of the following functional blocks:

- An order in which communication edges are considered for coloring.
- A \textit{ConflictFreeColor} function which determines the set of all existing colors which can be assigned to the edge under consideration without violating the problem constraints.
- A \textit{BestColor} rule to determine which conflict-free (non-conflicting) color to assign to the edge under consideration.
MaxSINRColor Function

\textbf{MaxSINRColor}(x)

\textbf{input}: \( \Phi(\cdot), G(\mathcal{V}, \mathcal{E}_c \cup \mathcal{E}_i) \)

\textbf{output}: A non-conflicting color

\( C \leftarrow \) set of existing colors

\( C_c \leftarrow \{ C(h) : h \in \mathcal{E}_c, h \text{ is colored, } x \text{ and } h \text{ have an edge conflict} \} \)

\( C_{cf} = C \setminus C_c \)

\textbf{if} \( C_{cf} \neq \emptyset \) \textbf{then}

\( r \leftarrow \) color in \( C_{cf} \) which results in maximum SINR at receiver of \( x \)

\( C(x) \leftarrow r \)

\textbf{if} \( \text{SINR}(x) > \gamma_c \) \textbf{then}

return \( r \)

\textbf{end if}

\textbf{end if}

return \( |C| + 1 \)
Labeler Function

Labeler($G_c$)

if $G_c$ is not empty then
    let $u$ be a vertex of $G_c$ of minimum degree
    $L(u) \leftarrow 1 + \text{Labeler}(G_c \setminus \{u\})$
    return $L(u)$
else
    return 0
end if

\begin{table}[h]
\begin{tabular}{c|c}
\hline
\text{Node} & \text{Label} \\
\hline
1 & 6 \\
2 & 4 \\
3 & 3 \\
4 & 5 \\
5 & 2 \\
6 & 1 \\
\hline
\end{tabular}
\end{table}
MaxSINRLinkSchedule Algorithm

MaxSINRLinkSchedule

**input:** $\Phi(\cdot), G(V, E_c \cup E_i)$

**output:** A coloring $C : E_c \rightarrow \{1, 2, \ldots\}$

$n \leftarrow \text{Labeler}(G_c)$

use successive breadth first searches to partition $G_c$ into oriented graphs $T_i, 1 \leq i \leq k$

**for** $i \leftarrow 1$ **to** $k$ **do**

**for** $j \leftarrow 1$ **to** $n$ **do**

**if** $T_i$ is out-oriented **then**

let $x = (s, d)$ be such that $L(d) = j$

**else**

let $x = (s, d)$ be such that $L(s) = j$

**end if**

$C(x) \leftarrow \text{MaxSINRColor}(x)$

**end for**

**end for**
FirstConflictFreeColor Function

FirstConflictFreeColor(x)

input: \( \Phi(\cdot), G_c(V, E_c) \)

output: A conflict-free color

\( C \leftarrow \) set of existing colors

\( C_c \leftarrow \{ C(h) : h \in E_c, h \text{ is colored, } x \text{ and } h \text{ have an edge conflict} \} \)

\( C_{cf} = C \setminus C_c \)

for \( i \leftarrow 1 \) to \( |C_{cf}| \) do

\( r \leftarrow i^{th} \) color in \( C_{cf} \)

\( E_i \leftarrow \{ h : h \in E_c, C(h) = r \} \)

\( C(x) \leftarrow r \)

if SINR at all receivers of \( E_r \cup \{x\} \) exceed \( \gamma_c \) then

return \( r \)

end if

end for

return \( |C| + 1 \)
ConflictFreeLinkSchedule Algorithm

**ConflictFreeLinkSchedule**

**input:** \( \Phi(\cdot), G_c(V, E_c) \)

**output:** A coloring \( C : E_c \rightarrow \{1, 2, \ldots\} \)

\( n \leftarrow \text{Labeler}(G_c) \)

use successive breadth first searches to partition \( G_c \) into oriented graphs \( T_i, 1 \leq i \leq k \)

**for** \( i \leftarrow 1 \) **to** \( k \) **do**

**for** \( j \leftarrow 1 \) **to** \( n \) **do**

**if** \( T_i \) is out-oriented **then**

let \( x = (s, d) \) be such that \( L(d) = j \)

**else**

let \( x = (s, d) \) be such that \( L(s) = j \)

**end if**

\( C(x) \leftarrow \text{FirstConflictFreeColor}(x) \)

**end for**

**end for**
RandomConflictFreeSchedule Algorithm

**RandomConflictFreeSchedule**

**input:** $\Phi(\cdot), G_c(\mathcal{V}, \mathcal{E}_c)$

**output:** A coloring $C : \mathcal{E}_c \rightarrow \{1, 2, \ldots\}$

$P =$ random permutation of communication edges in $\mathcal{E}_c$

for $i \leftarrow 1$ to $|\mathcal{E}_c|$ do

$\ x \leftarrow i^{th}$ edge in $P$

$\ C(x) \leftarrow \text{FirstConflictFreeColor}(x)$

end for
Simulation Model

Assumptions:

- Deployment region is a circular region of radius $R$.
- Location of every node is generated randomly, using a uniform distribution for its $X$ and $Y$ coordinates, in the deployment area. In polar coordinates, if $r_i = (R_i, \Theta_i)$, then $R_i^2 \sim U[0, R^2]$ and $\Theta_i \sim U[0, 2\pi]$.
- For a given $\Phi(N, (r_1, \ldots, r_N), P, \gamma_c, \gamma_i, \alpha, N_0)$, physical throughput and spatial reuse factor are computed by averaging these quantities over 1000 randomly generated networks.

We compare the performance of the following algorithms:

- Existing: ArboricalLinkSchedule, Truncated Graph-based Scheduling Algorithm
- Proposed: MaxSINRLinkSchedule, ConflictFreeLinkSchedule, RandomConflictFreeSchedule
Experiment 1

\[ R = 500 \text{ m}, \ P = 10 \text{ mW}, \ \alpha = 4, \ N_0 = -90 \text{ dBm}, \ \gamma_c = 20 \text{ dB} \text{ and } \gamma_i = 10 \text{ dB} \Rightarrow R_c = 100 \text{ m} \text{ and } R_i = 177.8 \text{ m}. \]
Experiment 2

\[ R = 700 \text{ m}, \; P = 15 \text{ mW}, \; \alpha = 4, \; N_0 = -85 \text{ dBm}, \; \gamma_c = 15 \text{ dB}, \; \gamma_i = 7 \text{ dB} \]

\[ R_c = 110.7 \text{ m} \quad \text{and} \quad R_i = 175.4 \text{ m}. \]
Notation

W.r.t. the communication graph \( G_c(V, E_c) \), let:

\[
\begin{align*}
    e & = \text{number of undirected edges} \\
    \nu & = \text{number of vertices} \\
    \rho & = \text{maximum degree (in-degree + out-degree) of any vertex} \\
    \theta & = \text{thickness of the graph}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Experiment 1</th>
<th>Experiment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu )</td>
<td>( \nu )</td>
</tr>
<tr>
<td>( \theta )</td>
<td>( \theta )</td>
</tr>
<tr>
<td>( \rho )</td>
<td>( \rho )</td>
</tr>
<tr>
<td>( e )</td>
<td>( e )</td>
</tr>
</tbody>
</table>

| 30 | 5 | 6 | 33 |
| 40 | 5 | 8 | 50 |
| 50 | 6 | 10 | 73 |
| 60 | 6 | 10 | 96 |
| 70 | 7 | 12 | 136 |
| 80 | 8 | 13 | 170 |
| 90 | 8 | 13 | 187 |
| 100 | 10 | 14 | 242 |
| 110 | 9 | 14 | 288 |

| 70 | 6 | 9 | 91 |
| 80 | 7 | 10 | 107 |
| 90 | 6 | 11 | 138 |
| 100 | 8 | 11 | 165 |
| 110 | 7 | 11 | 181 |
| 120 | 7 | 13 | 221 |
| 130 | 7 | 13 | 241 |
| 140 | 8 | 13 | 288 |
| 150 | 8 | 14 | 312 |

**Table:** Comparison of thickness, maximum degree and number of edges.
Theoretical Results

Theorem

*For an arbitrary graph $G$, the running time of MaxSINRLinkSchedule is $O(ev \log v + ev\theta)$.**
Theoretical Results

Theorem

For an arbitrary graph \( G \), the running time of MaxSINRLinkSchedule is \( O(\varepsilon v \log v + \varepsilon v \theta) \).

Proof.

1. Labeler function runs in time \( O(\varepsilon + v \log v) \) using a Fibonacci Heap.
2. Using Matroids, \( G_c \) can be partitioned into at most \( 6\theta \) oriented graphs in time \( O(\varepsilon v \log v) \). Thus, \( k \leq 6\theta \).
3. First oriented graph \( T_1 \) can be colored in time \( O(v^2) \). Any subsequent oriented graph \( T_j \), \( 2 \leq j \leq k \), can be colored in time \( O(\varepsilon v) \).
4. All oriented graphs can be colored in time \( O(\varepsilon v \theta) \).
Theoretical Results (cont’d)

Theorem

For an arbitrary graph $G$, the running time of ConflictFreeLinkSchedule is $O(ev \log v + ev\theta)$.

Theorem

For an arbitrary graph $G$, the running time of RandomConflictFreeSchedule is $O(e^2)$. 
Contributions of our work:

- Development of new SINR-based link scheduling algorithms for STDMA multihop wireless ad hoc networks under the physical interference model.
- Our algorithms achieve 25% higher physical throughput and 30% higher spatial reuse than existing link scheduling algorithms.
- Our algorithms have low computational complexity.

Future work:

- Application of stochastic techniques like simulated annealing and genetic algorithms to determine SINR-compliant STDMA link schedules.
Thank You