Algorithms for Multi-Channel Aggregated Convergecast in Sensor Networks

Amitabha Ghosh†  Ozlem D. Incel‡  V.S. Anil Kumar◊
Bhaskar Krishnamachari†
†Dept. of Electrical Engineering, University of Southern California
‡Dept. of Computer Science, University of Twente, Netherlands
◊Bio-Informatics Institute, Dept. of Computer Science, Virginia Tech

{amitabhg, bkrishna}@usc.edu, o.durmaz@cs.utwente.nl,
akumar@vbi.vt.edu
A Data Aggregation Network
A Data Aggregation Network
An Illustration

Setting

1. Half-duplex single transceiver
2. Nodes aggregate pkts from children and transmit only one pkt
3. TDMA
4. Interference causes pkt loss
5. Receiver-based frequency assignment
An Illustration

Setting

1. Half-duplex single transceiver
2. Nodes aggregate pkts from children and transmit only one pkt
3. TDMA
4. Interference causes pkt loss
5. Receiver-based frequency assignment
An Illustration

\[\begin{tikzpicture}
\node[circle,draw] (s) at (0,0) {$s$};
\node[circle,draw] (a) at (-3,-1) {$a$};
\node[circle,draw] (b) at (-1,-1) {$b$};
\node[circle,draw] (c) at (1,-1) {$c$};
\node[circle,draw] (d) at (-3,-2) {$d$};
\node[circle,draw] (e) at (-1,-2) {$e$};
\node[circle,draw] (f) at (1,-2) {$f$};
\node[circle,draw] (g) at (3,-2) {$g$};
\draw[dashed] (s) -- (a) node[midway,above] {$f_1$};
\draw[dashed] (s) -- (b) node[midway,above] {$f_1$};
\draw[dashed] (s) -- (c) node[midway,above] {$f_1$};
\draw[dashed] (a) -- (d) node[midway,above] {$f_1$};
\draw[dashed] (a) -- (e) node[midway,above] {$f_1$};
\draw[dashed] (b) -- (e);\draw (b) -- (f);\draw[dashed] (c) -- (f);\draw[dashed] (c) -- (g);\draw[dashed] (d) -- (g);\draw[dashed] (e) -- (g);\draw[dashed] (f) -- (g);
\draw (a) -- (b) node[midway,above] {5};\draw (b) -- (c) node[midway,above] {4};\draw (b) -- (d) node[midway,above] {6};\draw (c) -- (f) node[midway,above] {1};\draw (d) -- (e) node[midway,above] {3};\draw (d) -- (f) node[midway,above] {2};\draw (d) -- (g) node[midway,above] {1};\draw (e) -- (f) node[midway,above] {2};\draw (f) -- (g) node[midway,above] {1};%\end{tikzpicture}\]
Question

What is the fastest rate at which aggregated data can be collected from the network? (minimizing the schedule length)

<table>
<thead>
<tr>
<th>Receiver</th>
<th>Frame 1</th>
<th>Frame 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Slot 1</td>
<td>Slot 2</td>
</tr>
<tr>
<td>s</td>
<td>c</td>
<td>a,d</td>
</tr>
<tr>
<td>a</td>
<td>d</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>c</td>
<td>g</td>
<td></td>
</tr>
</tbody>
</table>
Minimum Frequency Assignment Problem

**MFAP**

Given a spanning tree $T$ on an arbitrary graph $G = (V, E)$, find the minimum number of frequencies that can be assigned to the receivers of $T$ such that all the interfering link constraints are removed.

**Figure:** Interfering edge structure

**Theorem**

*MFAP is NP-complete.*

*Proof: Reduction from Vertex Color.*
Lemma

Construct a constraint graph $G_C = (V_C, E_C)$ from $G = (V, E)$ as follows: For each receiver in $G$, create a vertex in $G_C$. Create an edge between two such vertices in $G_C$ if their corresponding receivers in $G$ are part of an interfering edge structure.

Then, the number of frequencies required to remove all the interfering link constraints is: $K_{\text{max}} \leq \Delta(G_C) + 1$, where $\Delta(G_C)$: max degree in $G_C$. 

![Diagram of constraint graph and BFS time slot assignment]
Largest Degree First (LDF)

1. while $V_C \neq \emptyset$ do
2. \hspace{1em} $u \leftarrow$ max degree in $V_C$
3. \hspace{1em} Assign the first available frequency to $u$ different from its neighbors
4. \hspace{1em} $V_C \leftarrow V_C \setminus \{u\}$
BFS Time Slot Assignment

Algorithm

Input: $T = (V, E_T)$
while $E_T \neq \emptyset$
    e ← next edge from $T$ in BFS order
    Assign minimum time slot to $e$
    $E_T \leftarrow E_T \setminus \{e\}$

Theorem

Algorithm $\text{BFS-TimeSlotAssignment}$ on a tree gives the minimum schedule length equal to $\Delta(T)$, where $\Delta(T)$ is the maximum node degree in $T$.

Proof by induction on $i$: $N(i + 1) = \max\{N(i), N(i) + 1\}$
Multiple-Frequency Minimum Time Scheduling Problem

**MFMTSP**

Given a spanning tree $T$ on an arbitrary graph $G = (V, E)$ and $q$ frequencies, find an assignment of frequencies and time slots such that the schedule length is minimized.

**Theorem**

*MFMTSP is NP-complete.*

*Proof: Reduction from Vertex Color.*
Reduction from Vertex Color.

\[ q(q-1)/2 \text{ links} \]

\[ u_{i1} \rightarrow u_{i2} \]

\[ v_i \rightarrow v_{i1} \rightarrow v_{i2} \]
Multiple-Frequency Minimum Time Scheduling Problem

Reduction from Vertex Color.

$q(q-1)/2$ links

$q^2$ interfering links
Multiple-Frequency Minimum Time Scheduling Problem

Reduction from Vertex Color.

\[ q(q-1)/2 \text{ links} \]

\[ v_i \rightarrow \]

\[ u_{i1} \quad u_{i2} \]

[Diagram of a tree with labeled vertices and time tags]
Multiple-Frequency Minimum Time Scheduling Problem

Reduction from Vertex Color.

[Diagram of a tree graph with nodes and edges labeled with frequencies and time durations.]

$q(q-1)/2$ links

$u_i1 \rightarrow u_i2$

$v_i \rightarrow v_i1 \rightarrow v_i2$

$T_b^1, T_b^2, T_b^3$
Lemma

Suppose \( \gamma_c \) denote the set of time slots to schedule the edges in cell \( c \). Then, the minimum schedule length \( \Gamma \) for the whole network is:

\[
\Gamma \leq 4 \cdot \max_c \{ |\gamma_c| \}, \forall \alpha \geq 2.
\]
Load-Balanced Frequency Assignment

\( R = \{v_1, \ldots, v_n\} \): receivers on \( T \)

\( m : R \rightarrow \{f_1, \ldots, f_K\} \)

If \( m(v_j) = f_k \), then children of \( v_j \) transmit on \( f_k \)

Define load on \( f_k \) under \( m \) as:

\[
l^m(f_k) = \sum_{m(v_j) = f_k} \deg^{in}(v_j)
\]

Then, a load-balanced frequency assignment \( m^* \) is:

\[
m^* = \arg \min_m \max_{f_k} \{l^m(f_k)\}
\]
Load-Balanced Frequency Assignment

\( R = \{v_1, \ldots, v_n\} \): receivers on \( T \)

\( m: R \to \{f_1, \ldots, f_K\} \)

If \( m(v_j) = f_k \), then children of \( v_j \) transmit on \( f_k \)

Define load on \( f_k \) under \( m \) as:

\[
l^m(f_k) = \sum_{m(v_j) = f_k} \text{deg}^{\text{in}}(v_j)
\]

Then, a load-balanced frequency assignment \( m^* \) is:

\[
m^* = \arg \min_m \max_{f_k} \{l^m(f_k)\}
\]

**Lemma**

Load-balanced frequency assignment is NP-complete.

\( \text{OPT (min-max) load is } L^{m^*} \).
**Algorithm** \textsc{FrequencyGreedy} ($\phi$)

(1) Sort the receivers in non-increasing order of in-degrees.
\[
\text{deg}^{\text{in}}(v_1) \geq \text{deg}^{\text{in}}(v_2) \geq \ldots \geq \text{deg}^{\text{in}}(v_n).
\]

(2) Starting from $v_1$, assign each successive node a frequency from \{\(f_1, \ldots, f_K\}\} that has the least load, breaking ties arbitrarily.

**Figure:** $l(f_1) = 5$, $l(f_2) = 5$
Algorithm **FrequencyGreedy** ($\phi$)

1. Sort the receivers in non-increasing order of in-degrees.
   
   \[ \operatorname{deg}_{\text{in}}(v_1) \geq \operatorname{deg}_{\text{in}}(v_2) \geq \ldots \geq \operatorname{deg}_{\text{in}}(v_n). \]

2. Starting from $v_1$, assign each successive node a frequency from \( \{f_1, \ldots, f_K\} \) that has the least load, breaking ties arbitrarily.

**Figure:** $l(f_1) = 5$, $l(f_2) = 5$

**Lemma**

*Algorithm FrequencyGreedy gives a \((4/3 - 1/3K)\) approximation on $L_{m^*}$.***

*Proof: Follows from Graham’s algorithm for scheduling jobs on identical parallel machines according to longest processing time first (LPT).*
An Upper Bound on $\gamma_c$

**Lemma**

If $L_c^\phi$ denote the load on the maximally loaded frequency in cell $c$ achieved by \textsc{FrequencyGreedy}, then any greedy time slot assignment can schedule all the edges in $c$ within $2L_c^\phi$ time slots, i.e.,

$$|\gamma_c| \leq 2L_c^\phi.$$
A Constant Factor Approximation

Theorem

Given a tree $T$ on a UDG $G$, and $K$ frequencies, there exists an algorithm $G$ that achieves a constant factor $8\mu_\alpha \left(\frac{4}{3} - \frac{1}{3K}\right)$ approximation on the optimal schedule length, where $\mu_\alpha > 0$ is a constant for a given cell size $\alpha \geq 2$, i.e.,

$$\Gamma_G = O(\Gamma_{OPT})$$
Proof Sketch

\( G \) consists of 2 phases:

1. Run **FrequencyGreedy** in each cell \( c_i \)
2. Run any greedy time slot assignment scheme
e.g., schedule a maximal number of edges in each slot

Lower bound on OPT:

\[
\Gamma_{OPT} \geq \max_c \left\{ \frac{L_{m}^{c^*}}{\mu} \right\}
\]

\[
\Gamma_{G} \leq 4 \cdot \max_c \{|\gamma_c|\} \leq 8 \cdot \max_c \{L_{c}^{\phi} \}
\]

\[
\leq 8 \cdot \max_c \{(4/3 - 1/3K) \cdot L_{c}^{m^*} \}
\]

\[
\leq 8\mu_c (4/3 - 1/3K) \cdot \Gamma_{OPT}
\]
Arbitrary Trees Under UDG

Figure: Shortest Path Tree

Figure: Minimum Spanning Tree
Theorem

Given a UDG $G$ and $K$ frequencies, there exists an algorithm $\mathcal{H}$ that achieves a constant factor $8\mu_\alpha \Delta_C$ approximation on the optimal schedule length, where $\mu_\alpha > 0$, $\Delta_C > 0$ are constants for a given cell size $\alpha \geq 2$, i.e.,

$$\Gamma_\mathcal{H} = O(\Gamma_{OPT})$$
Proof Sketch

\( V_c \): set of nodes in cell \( c \)
\( R_c = \{v_1, \ldots, v_n\} \): set of receivers in \( c \) on arbitrary \( T \)
\( \Delta^{in}(T) \): max in-degree in \( T \)

Lower bound on OPT:
\[
\Gamma_{OPT} \geq \max_c \left\{ \left\lceil \frac{|V_c|}{K} \right\rceil \right\} / \mu \alpha
\]

Then
\[
\Gamma_H \leq 4 \cdot \max_c \{|\gamma_c|\} \leq 8 \cdot \max_c \{L^\phi_c\}
\]
\[
\leq 8 \cdot \max_c \left\{ \left\lceil \frac{|V_c|}{K} \right\rceil \right\} \cdot \Delta^{in}(T)
\]
\[
\leq 8 \mu \alpha \Delta^{in}(T) \cdot \Gamma_{OPT}
\]

Bounded-degree spanning tree always exists on UDG. Therefore, proved.
Figure: Algorithm $G$ on Shortest Path Tree for Different Network Sizes
Figure: Algorithm $G$ on Minimum Interference Tree for Different Network Sizes
Conclusions

Summary

(1) Addressed a scheduling problem under aggregated convergecast using multiple channels.

(2) Proved a NP-completeness result on finding the minimum number of channels required to remove all the interfering links in an arbitrary wireless network.

(3) Proved a NP-completeness result on minimizing the schedule length for a given number of channels in an arbitrary wireless networks.

(4) Proposed an optimal time slot scheduling scheme when enough frequencies are available.

(5) Proposed constant factor approximation algorithms to minimize the schedule length on unit disk graphs.

(6) Evaluated algorithms using simulations: most of the times 3 to 4 frequencies are enough all the interfering links.