Cooperative Communications in Multi-hop Wireless Networks: Joint Flow Routing and Relay Node Assignment

Information Security Group
유재성
2010. 12. 09
1. Cooperative Communication (CC)
2. CC in multi-hop wireless network
3. Proposed Solution
4. Simulation Result
논문 연구 목적

1
논문 연구 목적

- Source Node
- Relay Node
- Destination Node

Session 1 (23 Mb/s)
Session 2 (32 Mb/s)
Session 3 (38 Mb/s)

Maximize Minimum Rate!
Cooperative Communication
2.1 Three Node CC Model

- Wireless Broadcast
- Relaying capability of other nodes so as to achieve higher throughput, lower transmission error
2.2 CC Mode

- Amplify and Forward (AF)
- Decode and Forward (DF)
- Direct Transmission (without CC)
CC in multi-hop wireless network
3.1 Network Setting

• Cooperative Relay (CR) operates at the physical layer while Multi-hop Relay (MR) operates at the network layer.
• A relay Node can serve as either CR or MR, but not both at the same time.
3.2 Mathematical Modeling

- **Source Nodes**  \( N_S = \{s_1, s_2, \ldots, s_{N_S}\} \)

- **Destination Nodes**  \( N_d = \{d_1, d_2, \ldots, d_{N_d}\} \)

- **Relay Nodes**  \( N_r = \{r_1, r_2, \ldots, d_{N_r}\} \)

- \( N = N_S + N_d + N_r = 2N_S + N_r \)
3.2 Mathematical Modeling

• Role of Relay Node
  – Binary variables
    \[ A_{uv}^w = \begin{cases} 
    1 & \text{If node } w \text{ is used as a CR on hop } (u, v), \\
    0 & \text{Otherwise.} 
  \end{cases} \]
    \[ B_{uv} = \begin{cases} 
    1 & \text{If } v \text{ is the next hop node of node } u, \\
    0 & \text{Otherwise.} 
  \end{cases} \]
  – For each relay node \( w \in N_r \), since it may be used as either a CR or MR, we can characterize this with the following two constraints.
    \[
    \sum_{u \in N} \sum_{v \neq w, u \neq v} A_{uv}^w + \sum_{t \in N} B_{tw} \leq 1 \quad (w \in N_r),
    \]
    \[
    \sum_{u \in N} \sum_{v \neq w, u \neq v} A_{uv}^w + \sum_{t \in N} B_{wt} \leq 1 \quad (w \in N_r).
    \]
3.2 Mathematical Modeling

- **Flow Routing**
  - \( f_{uv}(s_i) \) is a flow rate on link \((u,v)\) that is attributed to source-destination pair \((s_i,d_i)\).
  - The flow balance is formulated as follows
    \[
    \sum_{u \in N, u \neq d_i} f_{uw}(s_i) = \sum_{v \in N, v \neq s_i} f_{vw}(s_i)
    \]
    \[\quad (s_i \in N_s, w \in N, w \neq d_i, w \neq s_i).\]
  - It is easy to show that all the data generated by a source node, will reach its destination node.
    \[
    \sum_{w \in N, w \neq s_i} f_{siw}(s_i) = \sum_{w \in N, w \neq d_i} f_{wd_i}(s_i)
    \]
3.2 Mathematical Modeling

• Capacity
  - The aggregate flow rates traversing link \((u,v)\) must not exceed the capacity on link.

\[
\sum_{s_i \in N_s} f_{uv}(s_i) \leq \left(1 - \sum_{w \in N_r} A^w_{uv} \right) C_D(u,v) B_{uv} + \sum_{w \in N_r} A^w_{uv} C_{AF}(u,w,v) B_{uv}
\]

\((u \in N, v \in N, u \neq v)\).

  - If direct transmission is employed, then the first term on the right-hand-side(RHS) is non-zero and the second term is 0; the converse is true when CC is employed.
3.3 Problem Formulation

- The goal is to **maximize the minimum flow rate among all active sessions**
- **End-to-End flow rate (throughput) as** 
  \[ R_{s_i} = \sum_{v \in N} f_{s_i v}(s_i). \]
- \( R_{\text{min}} \) is a minimum flow rate among all sessions

  \[ R_{\text{min}} \leq \sum_{v \in N} f_{s_i v}(s_i) \quad (s_i \in N_s). \]

- The objective is to **maximize** \( R_{\text{min}} \)
3.3 Problem Formulation

- Reformulate into a linear constraint

\[
\sum_{s_i \in N_s} f_{uv}(s_i) \leq \left(1 - \sum_{w \neq u, w \neq v} A_{uv}^w\right) C_D(u, v) - B_{uv} + \sum_{w \in N_r} A_{uv}^w C_{AF}(u, w, v)
\]

\[
(u \in N, v \in N, u \neq v).
\]

- Property

\[
B_{uv} \cdot A_{uv}^w = A_{uv}^w \text{ for any } u \in N, v \in N, v \neq u, w \in N_r, w \neq u, w \neq v.
\]
### 3.3 Problem Formulation

\[
\begin{align*}
\text{Max} & \quad R_{\text{min}} \\
R_{\text{min}}, f_{uv}(s_i) & \geq 0 \quad (s_i \in N_s, u \in N, v \in N, u \neq v, u \neq d_i, v \neq s_i) \\
A^w_{uv}, B_{uv} & \in \{0, 1\} \quad (w \in N_r, u \in N, u \neq v \neq w)
\end{align*}
\]

- \(R_{\text{min}}, f_{uv}(s_i), A^w_{uv}, B_{uv}\) are optimization variables.

Mixed integer linear program (MILP)
Proposed Solution
4.1 Algorithm Overview

- Solving a relaxed version of the MILP
- FSC Algorithm
  - Phase 1: Path Determination
  - Phase 2: CR Assignment
  - Phase 3: Flow Re-calculating
- Generating a cutting Plane
- Selection of Branching Variable
4.2 Branch-and-cut

• For the MILP problem formulation, authors devised a solution procedure based on the branch-and-cut framework

• Branch-and-cut is an enhancement of branch-and-bound with the cutting plane method to deal with integer variables
4.2 Branch–and–cut

- Integer Programming Problem

\[
\begin{align*}
\text{min } z & := -6x_1 - 5x_2 \\
\text{subject to } & \quad 3x_1 + x_2 \leq 11 \quad \text{(3, 7)} \\
& \quad -x_1 + 2x_2 \leq 5 \\
& \quad x_1, x_2 \geq 0, \text{ integer.}
\end{align*}
\]
4.2 Branch-and-cut

- Integer Programming Sub-problem

\[
\begin{align*}
\text{min } & \quad z := -6x_1 - 5x_2 \\
\text{subject to } & \quad 3x_1 + x_2 \leq 11 \\
& \quad -x_1 + 2x_2 \leq 5 \\
& \quad x_1 \geq 3 \\
& \quad x_1, x_2 \geq 0, \text{ integer.}
\end{align*}
\]

\[
\begin{align*}
\text{min } & \quad z := -6x_1 - 5x_2 \\
\text{subject to } & \quad 3x_1 + x_2 \leq 11 \\
& \quad -x_1 + 2x_2 \leq 5 \\
& \quad x_1 \geq 2 \\
& \quad x_1, x_2 \geq 0, \text{ integer.}
\end{align*}
\]
4.2 Branch–and–cut

- Integer Programming Sub–problem

$$\begin{align*}
\text{min } z &= -6x_1 - 5x_2 \\
\text{subject to } &
\quad 3x_1 + x_2 \leq 11 \\
\quad -x_1 + 2x_2 \leq 5 \\
\quad x_1 \leq 2 \\
\quad 2x_1 + x_2 \leq 7 \\
\quad x_1, x_2 \geq 0, \text{ integer.}
\end{align*}$$

- Min $z = -28$
4.3 FSC Algorithm

- After solving the relaxed MILP, the solution may have fractional values for some of the \( A_{uv} \)’s or \( B_{uv} \)’s.
- The proposed FSC is a local search algorithm that constructs a feasible solution based on a given infeasible solution.

FSC Algorithm
- Phase 1 : Path Determination
- Phase 2 : CR Assignment
- Phase 3 : Flow Re-calculation
4.3 FSC Algorithm

- Phase 1: Path Determination
  - “Widest pipe” approach
  - Case 1 (Simple Path)
4.3 FSC Algorithm

- Phase 1: Path Determination
  - “Widest pipe” approach
  - Case 2 (Overlapping Path)
    When, encountered and intermediate node from the set of $N_s$ or $N_d$ (source or destination nodes)
4.3 FSC Algorithm

- Sub-Case 2.1: The encountered intermediate node is the source node of another session.
  - Encountered intermediate node ($s_j$) is included in the path as the next-hop node.
  - $s_j$ recorded in a special list (denoted as £)
4.3 FSC Algorithm

- Sub-Case 2.2: The encountered intermediate node is a destination node

Scenario A. (This node is the destination node of a source node in £)
4.3 FSC Algorithm

- Sub-Case 2.2: The encountered intermediate node is a destination node

Scenario B. (This node is the destination node of the current path under construction)
4.3 FSC Algorithm

- Sub-Case 2.2: The encountered intermediate node is a destination node

**Scenario C.** (This node is the destination node whose source node is not on the current path under construction)
4.3 FSC Algorithm

- Phase 2: CR Assignment
  - Introduce capacity-flow-ratio (CFR) for a hop as the ratio of the hop’s capacity to the number of overlapping sessions on that hop.
  - Sorting all paths in increasing order of CFR
  - The assignment of CRs start with the hop with the minimum CFR
  - In the case that the largest $A_{uv}^w$ is 0, no CR node will be assigned.
4.3 FSC Algorithm

- Phase 3: Flow Re-calculation
  - $A^w_{uv}$'s and $B_{uv}$'s are now fixed (either 0 or 1)
  - Original MILP is reduced to an LP with variables $R_{\text{min}}$ and $f_{uv}(s_i)$'s
  - Solve this LP and obtain a feasible solution for $f_{uv}(s_i)$'s
4.4 Generating a Cutting Plane

- Examine the values of $A_{uv}^w$’s and $B_{uv}$’s in the solution to a relaxed MILP problem

- If there are multiple fractional values, one of the fractional $A_{uv}^w$’s or $B_{uv}$’s will be chosen to generate a cutting plane

- Strategy: If any $A_{uv}^w$ or $B_{uv}$ is assigned to 1, then some other relevant variables can be assigned to 0. (Ex. $A_{uv}^w$ is assigned to 1, then node $w$ cannot be used as MR on any path)

- Can reduce the problem size
4.5 Selection of Branching Variable

- When the cutting plane is no longer able to offer much improvement in upper and lower bounds for a relaxed problem, move on to branching process
- Choose an $A_{uv}^w$ or $B_{uv}$ that is nearest to either 0 or 1
- If the variable that is closest to 0 is chosen, then in the two sub-problem, it will have the value 1 in one sub-problem and 0 in the other
- For the sub-problem with its value of 1, the new upper bound may be reduced significantly. It has potential of having new upper $< $ current lower bound
Simulation Result
5.1 Simulation with 20-node network

THROUGHPUT COMPARISON FOR THE 20-NODE NETWORK

<table>
<thead>
<tr>
<th>Session</th>
<th>With CC (Mb/s)</th>
<th>Without CC (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀ – d₀</td>
<td>36.1</td>
<td>29.2</td>
</tr>
<tr>
<td>s₁ – d₁</td>
<td>30.3</td>
<td>20.5</td>
</tr>
<tr>
<td>s₂ – d₂</td>
<td>32.3</td>
<td>22.0</td>
</tr>
<tr>
<td>s₃ – d₃</td>
<td>28.3</td>
<td>23.0</td>
</tr>
<tr>
<td>s₄ – d₄</td>
<td>32.2</td>
<td>27.3</td>
</tr>
</tbody>
</table>
5.2 Simulation with 25-node network

**Throughput Comparison for the 25-node Network**

<table>
<thead>
<tr>
<th>Session</th>
<th>With CC (Mb/s)</th>
<th>Without CC (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0 - d_0$</td>
<td>44.5</td>
<td>31.9</td>
</tr>
<tr>
<td>$s_1 - d_1$</td>
<td>47.4</td>
<td>34.8</td>
</tr>
<tr>
<td>$s_2 - d_2$</td>
<td>50.4</td>
<td>42.1</td>
</tr>
<tr>
<td>$s_3 - d_3$</td>
<td>49.0</td>
<td>31.1</td>
</tr>
<tr>
<td>$s_4 - d_4$</td>
<td>46.6</td>
<td>34.8</td>
</tr>
</tbody>
</table>
THANK YOU
SUNG KYUN KWAN UNIVERSITY