# Link Scheduling in Wireless Cooperative Communication Networks

Chenxi Qiu and Haiying Shen
Department of Electrical and Computer Engineering, Clemson University
({chenxiq, shenh}@clemson.edu)

## Background

Link scheduling: given a set of links, we need to determine which links should be active at what times.

Cooperative communication (CC): relay nodes storing the copies of messages are allowed to send the copies to the receiver simultaneously, and the receiver can combine the signal power of the received copies using cooperative diversity technique (e.g., maximal ratio combining (MRC)) to recover the message.

How to choose interference model for the link scheduling problem in CC?

1. Graph based model: define a set of interference edges to describe the conflicts among nodes.
2. Signal to interference plus noise ratio (SINR) model: a message is received successfully if only if the SINR is higher than a hardware-defined threshold (more suitable for the scheduling problem in CC)

Comparing to graph based model, the SINR model offers a more realistic representation of wireless networks. Accordingly, we study the cooperative link scheduling problem based on the SINR model.

## Problem statement

Given a finite set of nodes in a geometric plane, a set of requests, where each request is composed of a set of links and a receiver, and the decoding threshold.

**Decision version of CLS**

**Instance:** A finite set of nodes in a geometric plane $V$, a set of requests $F = \{f_1, \ldots, f_N\}$, where each request $f_i$ is composed of a set of links $l_i$ and a receiver $r_i$, and the decoding threshold $\gamma_{th}$.

**Question:** Existence of a CLS schedule s. t. each receiver $r_i$ can be informed by the given time constraint.

*We can prove CLS is NP-complete by reducing it from the Knapsack problem.*

## Design Details

Due to the hardness of the problem, we cannot find the optimal solution in polynomial time. Hence, we proposed a link length diversity (LLD) based algorithm (or LLD-CLS for short) for CLS.

Main idea of LLD-CLS :

1. Find the link with the shortest length for each receiver;
2. Build disjoint link classes according to the links' length.
3. For each link class, construct a feasible schedule using a greedy strategy.

In particular, in step 2: we build $g(K)$ disjoint link classes $L_1, \ldots, L_{g(K)}$ according to the links' lengths.

$$L_k = \{l \in L \mid 2^k\sigma \leq d(l) \leq 2^{k+1}\sigma\}$$

In step 3: when scheduling each link class $L_k$, the whole region is partitioned to a set of squares with size $\beta k = 2^{k+1}\sigma\beta$, where $\beta$ is specified by the length of the shortest link in $L_k$, and all the squares are colored regularly with 4 colors (see the figure below). Links whose receivers belong to different cells of the same color are scheduled simultaneously. Then, the distances between the links activated simultaneously are large enough, and hence the interference is overcome.

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 1 | 2 | 1 | 2 |
| 3 | 4 | 3 | 4 | 3 | 4 |
| 1 | 2 | 1 | 2 | 1 | 2 |
| 3 | 4 | 3 | 4 | 3 | 4 |
| 1 | 2 | 1 | 2 | 1 | 2 |
| 3 | 4 | 3 | 4 | 3 | 4 |

Partitioning and coloring

Theorem 1: Every receiver in LLD-CLS can successfully receive the packet.

Theorem 2: The approximation ratio of LLD-CLS is O(1).

## Experimental Results

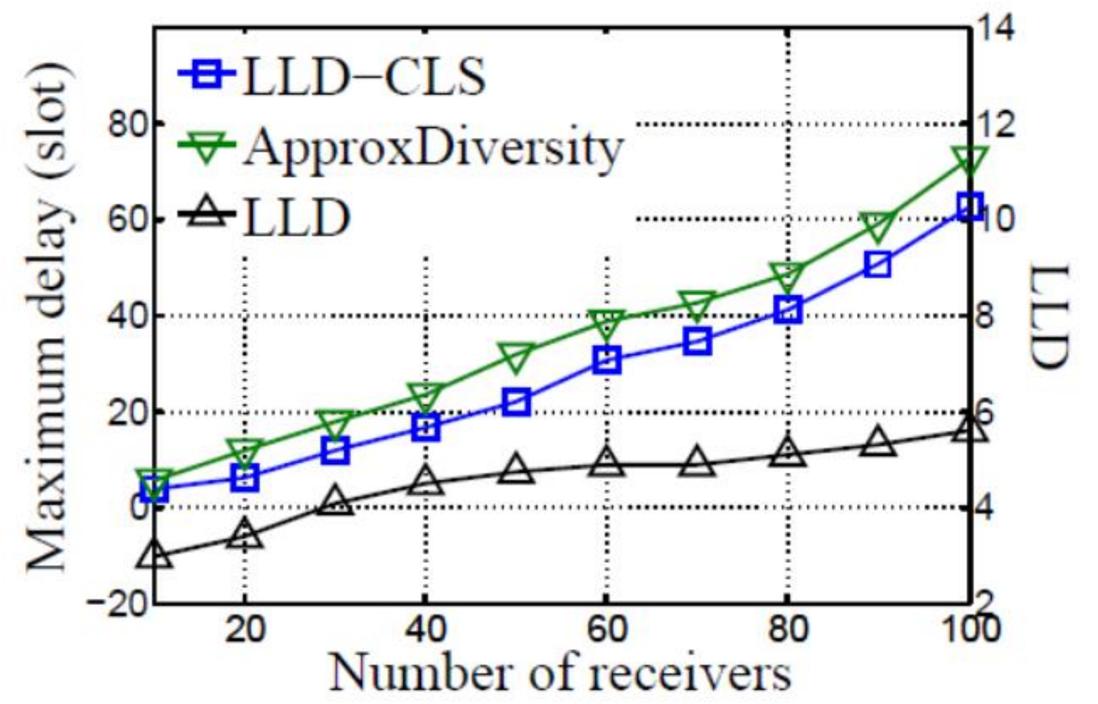
We conducted simulation using Matlab. All nodes were distributed uniformly at random on a plane field of size 100 ×100. The number of senders = 200 and the number of receivers from 10 to 100 with 10 increase in each step

Comparison metric:

Thoughput: the number of receivers informed in a single time slot.

Comparison methods:

ApproxDiversity, which partitions the link set into disjoint link classes and schedules the links in each class separately [1].




Throughput

Result: ApproxDiversity<LLD-CLS

References:
[1] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in Proc. of Mobihoc, 2007

## Future Work

We will take into account probabilistic fading models for the link scheduling problem.

## Acknowledgments

U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, and Microsoft Research Faculty Fellowship 8300751.