Rate Region of Scheduling a Wireless Network with Discrete Propagation Delays

Jun MA, Yanxiao Liu and Shenghao Yang

The Chinese University of Hong Kong, Shenzhen, Shenzhen, China.
Shenzhen Research Institute of Big Data, Shenzhen, China.

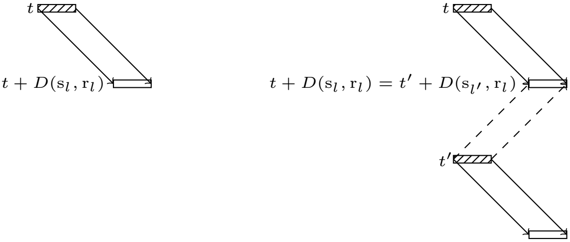
March 15, 2021

Motivation

- Wireless communication media, e.g., radio, light and sound, all have nonzero signal propagation delays.
- In underwater acoustic communications, the propagation delay can be longer than seconds.
- Previous researches show that taking propagation delay into consideration have signification advantage in throughput and energy consumption.

Network model

- Nodes are indexed by $1, 2, \ldots, N$.
- The signal propagation delay from i to j is $D(i, j) \in \mathbb{Z}^+$.
- A link l is a pair $(\mathrm{s}_l, \mathrm{s}_l)$. Denote $\mathcal{L}$ as a finite set of all the links.
- $\mathcal{I}(l)$ is the collision set of l. When l is active in timeslot t, a *collision occurs* if any $l' \in \mathcal{I}(l)$ is active in the timeslot $t + D(\mathrm{s}_l, \mathrm{r}_l) - D(\mathrm{s}_{l'}, \mathrm{r}_l)$.
- Let $\mathcal{I} = (\mathcal{I}(l), l \in \mathcal{L})$ be the *collision profile* of the network.
- Let $D_{\mathcal{L}}(l, l') = D(\mathrm{s}_l, \mathrm{r}_l) - D(\mathrm{s}_{l'}, \mathrm{r}_l)$ be the link-wise propagation delay



Network model as weighed directed graph

Our network model, denoted by $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$. The network $\mathcal{N}$ can be regarded as a weighted directed graph:

- The set of vertices is specified by $\mathcal{L}$;
- The set of edges is specified by $\mathcal{I}$;
- (l, l') is a directed edge if $l' \in \mathcal{I}(l)$, and has weight $D_{\mathcal{L}}(l, l')$.

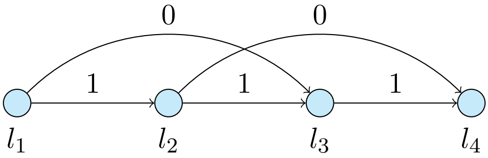
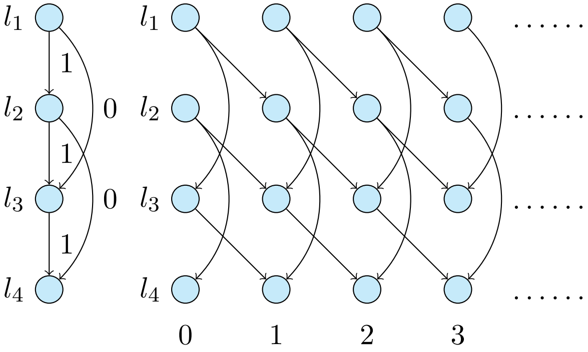


Figure: The graphical representation of $\mathcal{N}^{\text{line}}_{4,1}$.

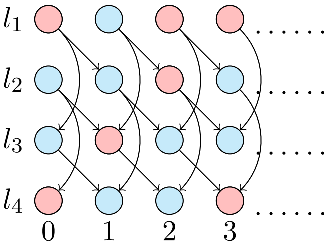
Network model and periodic graph

The *(directed) periodic graph* $\mathcal{N}^\infty$ induced by $\mathcal{N}$.



Periodic graph

A collision free schedule on $\mathcal{N}$ indicates an independent set of the *(directed) periodic graph* $\mathcal{N}^\infty$ induced by $\mathcal{N}$.



$$S = \begin{matrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{matrix} \begin{bmatrix} 1 & 0 & 1 & 1 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 1 & 0 & 0 & 1 & \dots \end{bmatrix}.$$

Rate region

- The collection $\mathcal{R}^{\mathcal{N}}$ of all the achievable rate vectors is called the *rate region* of $\mathcal{N}$.
- For a network $\mathcal{N}$, the rate region $\mathcal{R}^{\mathcal{N}}$ can be achieved using collision-free, periodic schedules only.
- Denote D^* as the maximum linkwise propagation delay.

Rate region by subgraphs

Define $\mathcal{N}^T$ as the subgraph of $\mathcal{N}^\infty$ with the vertex set $\mathcal{L} \times \{0, 1, \ldots, T-1\}$.
Define $\mathcal{R}^{\mathcal{N}^T}$ as the convex hull of the rate vectors of all the independent sets of $\mathcal{N}^T$.

Theorem

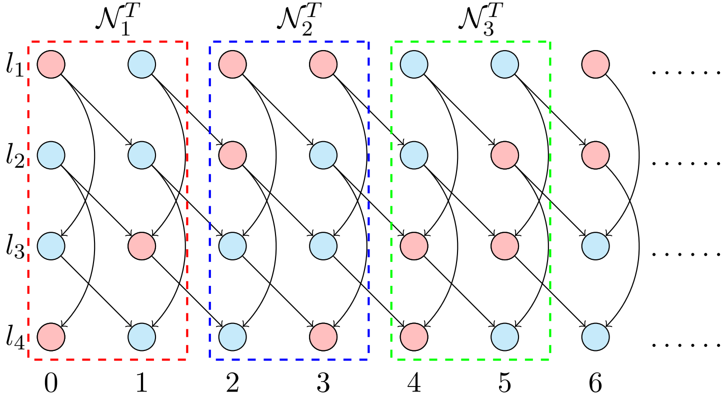
For a network $\mathcal{N}$,

$$\mathcal{R}^{\mathcal{N}} = \mathrm{closure}\left(\cup_{T=1,2,\ldots} \frac{T}{T+D^*} \mathcal{R}^{\mathcal{N}^T} \right),$$

where $\mathrm{closure}(\mathcal{A})$ is the closure of set A.

Conditional independence property

For subgraphs $\mathcal{N}_1^T$, $\mathcal{N}_2^T$ and $\mathcal{N}_3^T$, $\mathcal{N}_3^T$ is conditional independent with $\mathcal{N}_1^T$.
The selection of $\mathcal{N}_{i+1}^T$ is only determined by $\mathcal{N}_i^T$ when $T \geq D^*$.



Scheduling Graphs

A *scheduling graph* is a directed graph denoted by $(\mathcal{M}_T, \mathcal{E}_T)$ defined as follows:

- $\mathcal{M}_T$ is the collection of all independent sets of $\mathcal{N}^T$.
- $\mathcal{E}_T$ is the collection of all independent sets of $\mathcal{N}^{2T}$.

Scheduling Graphs

Example

For $\mathcal{N}_{4,1}^{\text{line}}$, $(\mathcal{M}_1, \mathcal{E}_1)$ has the vertex set $\mathcal{M}_1 = \{v_i, i = 0, 1, \ldots, 8\}$ where

$$v_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, v_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$v_5 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, v_6 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, v_7 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, v_8 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

Scheduling Graphs

Example

The adjacency matrix

$$
\begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}
\end{array}.
$$

Scheduling Graphs and schedules

Theorem

A collision-free schedule S, when $T \geq D^*$ can be represented by a directed path in a schedule $(\mathcal{M}_T, \mathcal{E}_T)$.

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_0	1	1	1	1	1	1	1	1	1
v_1	1	1	0	1	1	1	0	0	1
v_2	1	1	1	0	1	1	0	0	1
v_3	1	1	1	1	0	0	1	1	0
v_4	1	1	1	1	1	1	1	1	1
v_5	1	1	0	1	1	1	0	0	1
v_6	1	1	0	0	1	1	0	0	0
v_7	1	1	1	0	0	1	0	0	0
v_8	1	1	1	1	0	1	0	1	0

$$S = \begin{matrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{matrix} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Rate Region by Cycles

- A collision-free schedule S of period forms a closed path in $(\mathcal{M}_T, \mathcal{E}_T)$.
- A closed path can be decomposed into a sequence of (not necessarily distinct) cycles.

Rate Region by Cycles

For a finite directed graph $\mathcal{G}$, we know that $\mathrm{cycle}(\mathcal{G})$ is finite. Define

$$\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_T)} = \mathrm{conv}(\{R_C : C \in \mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_T)\}),$$

As $(\mathcal{M}_T, \mathcal{E}_T)$ is finite, $\mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_T)$ is finite and hence $\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_T)}$ is a closed set.

Theorem

For a network $\mathcal{N}$ and any integer $T \geq D^*$, $\mathcal{R}^{\mathcal{N}} = \mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_T)}$.

Dominance properties

- $A \preccurlyeq B$ if all the entries of A are not larger than the corresponding entries of B at the same positions.
- For a set $\mathcal{A}$ with partial order $\succcurlyeq$, we write $\max_{\succcurlyeq} \mathcal{A}$ as the smallest subset $\mathcal{B}$ of $\mathcal{A}$ such that any element of $\mathcal{A}$ is dominated by certain elements of $\mathcal{B}$.

Algorithms for Rate Region: Straightforward approach

- Instead of $\mathcal{E}_T$, we can find $\mathcal{E}^* = \max_{\succcurlyeq} \mathcal{E}_T$, which is the collection of maximal independent sets of $\mathcal{N}^{2T}$.
- Using the Bron–Kerbosch algorithm to enumerate all the maximal independent sets of $\mathcal{N}^{2T}$ to calculate $\mathcal{M}_T$ and $\mathcal{E}^*$, where T can be as small as D^*.
- Using a backtracking algorithm to enumerate all the cycles in $(\mathcal{M}_T, \mathcal{E}_T)$.

An Incremental Approach for Maximal Cycles

- Algorithm 1: Enumerating the maximal paths incrementally.
- Algorithm 2: Finding all cycles dominated by a path.

Algorithm 1: enumerating the maximal paths incrementally.

- $\mathcal{G}_1 = (\mathcal{M}_L^*, \mathcal{U}_0', \mathcal{M}_R^*)$, let $\mathcal{M}_L^*$ (resp. $\mathcal{M}_R^*$) be the collection of B such that $(B, B') \in \mathcal{E}^*$ (resp. $(B', B) \in \mathcal{E}^*$) for certain B'. $\mathcal{E}^* \subset \mathcal{M}_L^* \times \mathcal{M}_R^*$.
- For $k > 1$, we define a directed $(k+1)$-partite graph $\mathcal{G}_k$.

$$\mathcal{G}_k = (\mathcal{M}_L^*, \mathcal{U}_0, \mathcal{V}, \mathcal{U}_1, \mathcal{V}, \ldots, \mathcal{U}_{k-2}, \mathcal{V}, \mathcal{U}_{k-1}', \mathcal{M}_R^*),$$

  where $\mathcal{V} = \{B \wedge B' : B \in \mathcal{M}_R^*, B' \in \mathcal{M}_L^*\}$. $\mathcal{G}_{i+1}$ can be calculated using $\mathcal{G}_i$ and $\mathcal{E}^*$.
- Any k-length maximal path in $(\mathcal{M}_T, \mathcal{E}_T)$ is a path of length k in $\mathcal{G}_k$.

Algorithm 1: example

Example

For the scheduling graph $(\mathcal{M}_1, \mathcal{E}_1)$ of $\mathcal{N}_{4,1}^{\text{line}}$, $\mathcal{G}_1 = (\mathcal{M}_L^*, \mathcal{E}^*, \mathcal{M}_L^*)$ is characterized in Example 2, where $\mathcal{M}_L^* = \mathcal{M}_R^* = \{v_5, v_6, v_7, v_8\}$. For this example, we have $\mathcal{V} = \mathcal{M}_1$, $\mathcal{U}_0$ has an adjacent matrix

$$\begin{array}{c} \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

Algorithm 1: example

Example

$\mathcal{U}_1'$ and $\mathcal{U}_2'$ have, respectively, the adjacent matrices

$$\begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array} \begin{array}{c} \begin{array}{cccc} v_5 & v_6 & v_7 & v_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array} \text{ and } \begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array} \begin{array}{c} \begin{array}{cccc} v_5 & v_6 & v_7 & v_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

Algorithm 1: example

Example

Moreover, we have the adjacent matrix of $\tilde{\mathcal{U}}'$:

$$
\begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array}
\begin{array}{c}
\begin{array}{cccc} v_5 & v_6 & v_7 & v_8 \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0
\end{bmatrix}
\end{array}.
$$

From the adjacent matrices, we see that $\mathcal{U}_1', \mathcal{U}_2' \subset \tilde{\mathcal{U}}'$.

Algorithm 1

Example

Consider the network $\mathcal{N}_{4,1}^{\text{line}}$. For $k = 1, \dots, 4$, we list the number of paths in $\mathcal{G}_k$ and the total number of length-k paths in $(\mathcal{M}_1, \mathcal{E}_1)$ in the following table:

	$k=1$	$k=2$	$k=3$	$k=4$
Number of length-k paths in $\mathcal{G}_k$	6	16	64	180
Number of length-k paths in $(\mathcal{M}_1, \mathcal{E}_1)$	56	363	2357	152633

Algorithm 2: Finding all cycles dominated by a path.

Concluding remarks

- The essential problem: independent sets of a periodic graph.
- Connect periodical independent sets with cycles in scheduling graphs.
- Simplifying the computation costs of enumerating cycles by exploring dominance property.