

CATBSにおける媒介中心性に基づいた 貪欲なスケジューリングアルゴリズム

齊藤 明日香¹ 吉廣 卓哉²

概要: 本論文では, CATBS におけるスケジューリング問題を, より小さいメモリ消費量で解くスケジューリングアルゴリズムを提案する. CATBS とは, スロット化 CSMA を用いた無線メッシュ網の実現方式の一つであり, CSMA により近隣ノード間のフレーム衝突を解決する一方で, スケジューリングにより隠れ端末の関係にあるノード間の衝突を回避することで, ネットワーク全体でフレーム衝突のない効率的な通信を実現する. しかし, CATBS におけるスケジューリングアルゴリズムはネットワーク内の端末数に対して指数的にメモリ消費量が増大する問題がある. 本研究では, 媒介中心性に基づいた貪欲なスケジューリングアルゴリズムを提案し, 現実的な小さいメモリ消費量で CATBS におけるスケジューリング問題を解く. 評価を通じて, 提案アルゴリズムにより, 小さいメモリ消費量と短い計算時間で, CATBS に近い精度の解を得られることを示した.

1. はじめに

近年, 無線メッシュ網 [1] の研究が注目されている. 無線メッシュ網とは, 固定された無線基地局により構成されたマルチホップネットワークである. 無線メッシュ網では無線通信によるために端末間で電波が衝突することが問題となっており, これを防ぐために IEEE802.11[2] に採用されているアクセス制御方式である CSMA が用いられることが多い. しかし, CSMA を用いる場合に隠れ端末問題の影響が大きく, 通信フレームが頻繁に衝突することで, 通信速度が十分に上がらない問題がある.

そこで, 我々の研究グループでは CATBS[3] を提案している. CATBS ではスロット化 CSMA を用いて, 隠れ端末問題による電波干渉を削減する. スロット化 CSMA とは, 単一周波数を一定時間ごとに区切ったスロットを用いることで, 仮想的にチャンネルを複数あるように見せる手法である. CATBS では隠れ端末問題の関係にある端末同士に異なるスロットを割り当てることで, 通信のタイミングをずらして電波干渉を回避する. しかし, CATBS におけるスケジューリングアルゴリズムは部分 MAX-SAT に帰着させて解くために, ネットワーク内の端末数に対して指数的にメモリ消費量が増大することが問題である.

本研究では, CATBS におけるスケジューリング問題を

メモリ消費を抑えて解くスケジューリングアルゴリズムを提案する. 具体的には, CATBS におけるスケジューリング問題を媒介中心性 [4] を用いた貪欲法に基づいて解く. 媒介中心性はネットワーク分析においてよく使われる中心性の指標の一つで, あるリンクがネットワーク内でどのくらい最短経路としてよく使われるかを表す指標である. 本研究では, この媒介中心性の値が高いリンクを残して通信経路を制限する.

本論文は以下のように構成される. 第2章では関連研究と, 提案手法の基礎と干渉を削減するスケジューリング法である CATBS について述べる. 第3章では本論文の提案手法である, CATBS のスケジューリング問題において, メモリ消費を抑えたスケジューリングアルゴリズムについて述べる. 第4章では, 提案手法について評価し, メモリ消費を抑えて CATBS に近い精度の解が得られることを示す. 最後に第5章で本論文をまとめる.

2. CATBS のスケジューリング問題

2.1 CATBS の概要

CATBS はスロット化 CSMA を用いることで, CSMA により近隣ノード間の電波干渉を回避し, スケジューリングにより隠れ端末問題の関係にあるノード間の電波干渉を回避することで, ネットワーク全体の電波干渉を削減する手法である. スロット化 CSMA は, 単一周波数チャンネルを複数のスロットに分割し, 各スロット内で CSMA を動作させる MAC プロトコルである. 隠れ端末問題の関係

¹ 和歌山大学大学院 システム工学研究科
和歌山市栄谷 930 番地

² 和歌山大学 システム工学部
和歌山市栄谷 930 番地



図 1: スロット化 CSMA

にある端末同士の通信に異なるスロットを割り当てることで、隠れ端末問題による電波干渉を最小化する。

2.2 スロット化 CSMA

スロット化 CSMA とは単一周波数チャンネルを一定時間ごとに複数に分割し、分割した各時間一つ一つをチャンネルとみなすことで、仮想的にチャンネルが複数あるように見せる手法である。区切られた時間一つ分をスロットと呼び、各スロット内では CSMA が働く。

スロット数を k とすると、各スロットは $1, 2, \dots, k, 1, 2, \dots$ のように識別する番号が与えられ、順に切り替えが繰り返されている。端末は決められたスロット時間のときのみフレームを送信する。異なるスロット間で送信時間が異なるため、隠れ端末問題の関係にあるリンクペアに異なるスロットを割り当てることで、干渉を防ぐことができる。

図 1 に例を示す。端末 A, B, C, D が図 1 のように配置されており、一つ隣の端末までは電波が届くものとする。このとき、A と B は同じスロット 1 を使って通信を開始するが、お互いの電波を検知できるため CSMA により干渉を回避できる。しかし、A と C は隠れ端末問題の関係にあるため、電波を検知することができない。A と C の通信を異なるスロットで行うように割り当てることにより、送信タイミングをずらし干渉を避けることができる。

2.3 CATBS におけるスケジューリング問題

2.3.1 有向グラフによるネットワークのモデル化

CATBS では干渉を削減するスケジューリング問題を定式化するために、無線メッシュ網を有向グラフで表す。ネットワークを $G = (V, E, C)$ と表し、頂点集合を V 、リンク集合を E 、スロット集合を C とする。頂点 u から v までをスロット c を使って張られたリンクを $e = (u, v, c)$ と定義する。ネットワーク G における u から v までの最短距離を $D_{u,v}^G$ と表す。リンク e_1 が e_2 の通信を妨害するとき、 e_1 と e_2 は妨害関係にあると定義する。ネットワーク G 内の e_1, e_2 のように妨害関係にあるリンクペアの集合を S_G と表し、 $|S_G|$ を衝突度と定義する。CATBS のスケジューリング問題では干渉リンクペア数を最小化すること、つまりこの衝突度を最小化することを目的としている。

CATBS では先に述べたように、スロット化 CSMA を用いるため、電波を検知できる頂点間は CSMA によって干渉が避けられる。ゆえに CATBS では、衝突度を隠れ端末問題の関係にあるリンクペアの数で定義する。

2 本のリンク $e_1, e_2 \in E$ を $e_1 = (u_1, v_1, c_1)$ と $e_2 =$

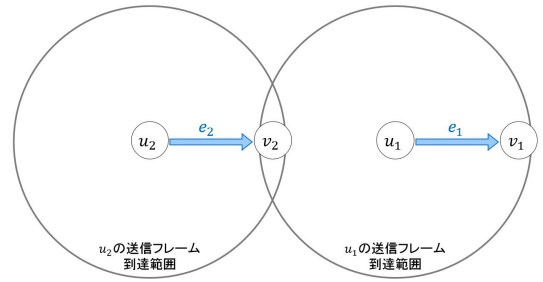


図 2: データフレーム同士の衝突例

(u_2, v_2, c_v) と表し、 e_1 と e_2 が妨害関係であるときを考える。 e_1 が e_2 を妨害する条件パターンは二種類ある。まず、条件 1 として、 e_1 と e_2 のデータフレーム同士が衝突するときに定義する。二つのリンクのスロットが同じであるとき、 u_1 が u_2 と隣接せず、さらに u_1 が v_2 と隣接するとき、二つのリンクは妨害関係にある。これは送信ノード同士がお互いの電波が届かないために、CSMA が機能せずフレームを待機することなく送信してしまうことで互いの送信フレームが衝突する。形式的な記述を以下に記す。また、条件 1 にあてはまる衝突の例を図 2 に示す。

条件 1: データフレーム同士の衝突

- (1) $c_1 = c_2$
- (2) $(u_1, u_2, c_1) \notin E$ (つまり、 u_1 と u_2 が隣接しない。)
- (3) $(u_1, v_2, c_1) \in E$ (つまり、 u_1 と v_2 が隣接する。)

次に、条件 2 として、 e_1 の Ack フレームと e_2 のデータフレームが衝突するときに定義する。二つのリンクのスロットが同じであるとき、 v_1 が u_2 と隣接せず、さらに v_1 が v_2 と隣接するとき、二つのリンクは妨害関係にある。これは u_1 が送ったデータフレームに対して、 v_1 が Ack フレームを返すとき、 v_1 と u_2 はお互いの電波が届かない範囲にあるために、送信を待機することなく開始してしまうことで、 v_1 の Ack フレームと u_2 の送信フレームが衝突する現象である。形式的な記述を以下に記す。また、条件 2 にあてはまる衝突の例を図 3 に示す。

条件 2: データフレームと Ack フレームの衝突

- (1) $c_1 = c_2$
- (2) $(u_1, u_2, c_1) \notin E$ (u_1 と u_2 が隣接しない。)
- (3) $(v_1, v_2, c_1) \in E$ (v_1 と v_2 が隣接する。)

2.3.2 CATBS のスケジューリング問題

本節では CATBS におけるスケジューリング問題の定式化について説明する。CATBS では衝突度を最小化するために、各リンクにデータフレームを送信できるスロットを割り当てる。さらに衝突度を減らすために、通信経路として使用可能なリンクを制限する。スケジューリング問題で

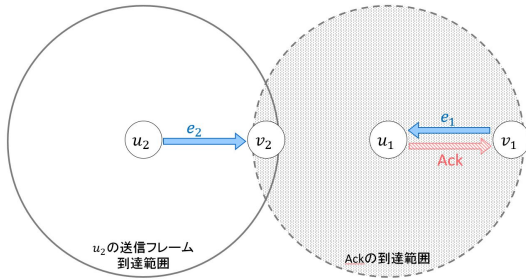


図 3: データフレームと Ack フレームの衝突例

は、入力されるグラフにおいて、使用可能なリンクはすべてのスロットでつながれているものとする。また、入力グラフから、衝突度を最小化するようなリンクを選んだ部分グラフを出力グラフとする。

グラフ内の使用可能なリンクを制限することによって、グラフ内の二頂点間の最短経路長が増加することが考えられる。最短距離が増加すると、フレーム送信において遅延が発生し、通信効率が低下する。CATBS では隠れ端末問題を解決するために、 k ホップまでの最短経路長の増加を許容する。CATBS におけるスケジューリング問題では制約として、値 k を用いて入力グラフの最短経路長と比べて k 以内の増加は許容することとする。また、各頂点は送信キューを一つだけ持つことを想定するため、一つの頂点ができる送信スロットは 1 つのみとする。送信キューに対して、スロットを複数持つとすると、送信キューの先頭で待機するフレームのスロットより、次に待機するスロットの方が時間的に早いスロットで送られるとき、先頭にあるフレームが残るために送信できない。これにより、通信効率が低下すると考えられるためである。

この問題を定式化する。入力を有向グラフ $G = (V, E, C)$ とし、出力を $G' = (V, E', C)$ とする。ここで、グラフ G 上においてノード u からノード v までの最短経路長を $D_{u,v}^G$ で表す。制約は二つ定義する。まず、出力グラフにおける各二頂点間の最短経路長は入力グラフにおける各二頂点間の最短経路長と比べて k までの増加とする。次に、一つのノードが使用する送信スロットは一つのみとする。定式化したスケジューリング問題は以下である。

スケジューリング問題の定式化

- 入力：有向グラフ $G = (V, E, C)$ 、経路長の増加許容量 k
- 出力： G の部分グラフ $G' = (V, E', C)$, ($E' \subset E$)
- 制約： $D_{u,v}^{G'} - D_{u,v}^G \leq k$,
1 つのノードが使用する送信スロットは 1 つのみ
- 目的関数： G' の衝突度 $|S_{G'}|$ を最小化する

2.3.3 CATBS におけるスケジューリングアルゴリズム

本節ではスケジューリング問題の解法アルゴリズムに部

分 MAX-SAT[6] を用いる。CATBS におけるスケジューリング問題は NP 困難な問題であるため、近似解を高速に求める必要がある。CATBS では、スケジューリング問題を部分 MAX-SAT に帰着して解く。部分 MAX-SAT は NP 困難な最適化問題であり、近年でも、高速にいい近似解を求めるソルバーのコンテストが開催されるなど、優れたソルバーのコンテストが開催されるなど、優れたソルバーが開発されており、CATBS はそれらを用いた高速計算を試みる。

2.4 問題点

CATBS ではスケジューリング問題を部分 MAX-SAT を用いて解くことで、近似解を高速に求める。しかし、部分 MAX-SAT はノード数すなわち無線ネットワーク内の端末数が増えるほど、部分 MAX-SAT の入力である論理式の節の数が指数的に増えるため、多くのメモリ量を必要とする。無線メッシュ網では無線ネットワークの拡張を目指しているため、ノード数が膨大になることが予想される。ゆえに、ネットワークが大きくなるほど、メモリ消費も膨大になると考えられる。また、無線メッシュ網は固定基地局によるネットワークであるため、ネットワークのトポロジは品番には変化しないが、電波状況やノードの故障等の要因のために、一時的にトポロジが変化する状況が考えられる。CATBS ではリンクを削減しているために、小さなトポロジ変化により端末間の経路長が増加したり孤立する端末ができることがあり得る。ゆえに、トポロジ変化に合わせてスケジューリングの再計算が必要であり、メモリ消費が大きいことは CATBS の問題点である。

本論文では、CATBS におけるスケジューリング問題を貪欲法に基づいて解くことで、メモリ消費の小さいスケジューリングアルゴリズムを提案する。貪欲法は計算の各段階で最も数値が良い解を選んでいき、ある評価基準のもとで最も良い解を見つけるためのアルゴリズムである。貪欲法による解法は高速で、またメモリ消費が小さいため、提案手法では CATBS におけるスケジューリング問題を貪欲法を用いて解いていく。

3. 提案するスケジューリングアルゴリズム

3.1 媒介中心性

本節では、本研究で用いる媒介中心性について説明する。ネットワークにおける中心性とは、ネットワーク内で、どの頂点またはリンクが要であるかを表すための指標である。中心性は、ネットワーク分析においてよく使われている指標で、複雑ネットワークの分野でもよく応用されている。媒介中心性は中心性の中でよく使われる指標の一つであり、ある頂点やリンクが、あらゆる二頂点間の最短経路のうち何本に含まれているかを表す。

ある頂点またはリンク i の媒介中心性を求めるとき、まずすべての二頂点間の経路のうち、何本の経路上にその頂点またはリンク i があるかを計算する。この値を媒介数とよび、 $b_i = \sum_{j \neq k} g_{jk}^i / g_{jk}$ で定義される。ただし、 g_{jk} は頂点 j から頂点 k までの最短経路の数を表し、 g_{jk}^i は j から k への最短経路のうち、頂点またはリンク i を通る経路の数を表す。さらに、求めた媒介数をグラフ上に存在する経路数で割ることで、ネットワーク内に存在する経路のうちどれくらいの割合を占めるかを $0 \sim 1$ の値で表す。これを媒介中心性とよび、 $C_{b_i} = 2b_i / (n-1)(n-2)$ で表す。ただし、 n はネットワーク内の頂点数を表す。

3.2 アイデア

本研究では CATBS におけるスケジューリング問題を二つに分割し、それぞれの問題を貪欲法に基づいたアルゴリズムによって解く。本研究の目的は、CATBS におけるスケジューリング問題をメモリ消費を抑えて解くことである。これを実現するために、本研究では媒介中心性を用いた貪欲法に基づくスケジューリングアルゴリズムを提案する。媒介中心性とは前節で述べたように、ある頂点やリンクが、すべての二頂点間の最短経路のうち何本に含まれるかを表しており、すべての二頂点間の最短経路のうちどれくらい経路としてよく使われているかを測ることができる。リンクを一本取り除くことにより、二頂点間の最短経路長が伸びることがある。媒介中心性の高いリンクは経路としてよく使われているために、そのリンクを取り除いたときのネットワーク全体に対する影響が大きくなる。提案アルゴリズムでは媒介中心性の高いリンクを残して、グラフ内のリンク数を最小化することで、グラフ内の衝突度を最小化するスケジュールを求める。

ここで CATBS ではリンクを制限することと、リンクにスロットを割り当てることによって衝突度を削減する。これに対して本研究では、媒介数を用いてリンクを削減すると同時にスロットの割り当てを行うと処理が複雑になると考えたため、リンクの削減をリンク削減問題として、スロットの割り当てをスロット割り当て問題としてそれぞれを独立して定式化し、それぞれの問題を解くアルゴリズムを与える。

リンク削減問題では媒介中心性を用いて、グラフ内のリンク数を最小化する。CATBS におけるスケジューリング問題と同様に、グラフからリンクを取り除いていくことで頂点間の最短経路長が伸びることがある。ここではリンク数を減らすことを優先とするため、最短経路長の増加が大きくなりすぎないように k ホップ以内であることを制約とすることで増加を抑える。定式化したリンク削減問題を以下に示す。

リンク削減問題

- 入力: 有向グラフ $G = (V, E)$, 経路長の増加許容量 k
- 出力: G の部分グラフ $G' = (V, E')$, ($E' \subset E$)
- 制約: $D_{u,v}^{G'} - D_{u,v}^G \leq k$
- 目的関数: G' のリンク数 $|E'|$ の最小化

スロット割り当て問題では衝突度を用いて、リンク削減問題によって出力されたグラフから、グラフ内の衝突度 $|S_{G'}|$ が小さくなるような、各リンクのスロットの割り当てを出力する。CATBS におけるスケジューリング問題と同様に、各ノードは送信キューを一つずつ持つことを想定するため、各頂点が使用できる送信スロットは一つだけに限ることを制約とする。これを満たすために、提案手法のスロット割り当て問題では、ノードごとにスロットを割り当て、そのノードから出るリンクはすべて、ノードと同じスロットを割り当てるものとする。定式化したスロット割り当て問題を以下に示す。

スロット割り当て問題

- 入力: リンク削減問題の出力グラフ $G' = (V, E')$, スロット集合 S
- 出力: 各リンクに対するスロット割り当て $f: E' \rightarrow S$
- 制約: 1つのノードが使用する送信スロットは1つ
- 目的関数: G' の衝突度 $|S_{G'}|$ の最小化

3.3 媒介中心性を用いたリンク削減問題の解法

本節では、リンク削減問題を解く提案アルゴリズム DeleteLink を説明する。リンク削減問題の提案アルゴリズムは媒介中心性を用いて、経路として重要度の低いリンクから取り除いていくことで、リンク数を最小化する。ここで、媒介中心性はある頂点またはあるリンクが、すべての二頂点間の最短経路数のうち何本を通るかを $0 \sim 1$ の値で割合を表しており、リンクの選び方に変化は出ないので、計算処理を減らすために本アルゴリズムでは媒介中心性を求める代わりに媒介数を計算する。

本アルゴリズムの手順を説明する。手順を Algorithm 1 に示す。まず、グラフ内のリンクをキューに入れておく。このキュー内で媒介数をもっとも小さいリンクを選び、そのリンクをグラフから取り除いてすべての二頂点間の最短経路長を計算し、入力グラフのすべての二頂点間の最短経路長と比べて k 以内の増加であればリンクを取り除いてグラフを更新する。この操作をキュー内のリンク候補がなくなるまで繰り返す。

提案アルゴリズムを具体的に説明する。提案アルゴリズムは入力にグラフ $G = (V, E)$ を与え、入力グラフの部分グラフであるリンク数を最小化したグラフ $G' = (V, E')$ を出力する。手順を一行ずつ説明する。1~2行目は初期化を

Algorithm 1: DeleteLink

Input: $G = (V, E)$
Output: $G' = (V, E')$

```
1   $Q \leftarrow E$ 
2   $G' \leftarrow G$ 
3   $BetweenCentrality(G)$ 
4  while ( $Q \neq \phi$ )
5       $e \leftarrow ChooseLink(Q)$ 
6       $G_{tmp} \leftarrow G' - e$ 
7       $BetweenCentrality(G_{tmp})$ 
8      if ( $D_{u,v}^{G_{tmp}} \leq D_{u,v}^G + k$ )
9           $G' \leftarrow G_{tmp}$ 
10     end if
11 end while
```

行う。リンク集合 E に含まれるすべてのリンクをキュー Q に入れ、2行目で入力グラフ G をプログラム内で操作するために G' にコピーする。3行目でグラフ G を引数として関数 $BetweenCentrality$ を呼び出し、グラフ内のすべての二頂点間の最短経路および最短経路長 $D_{u,v}^G$ を計算する。また最短経路長が同じ経路の数を経路数としていくつあるかを計算する。媒介数の計算方法については、Brandes のアルゴリズム [5] を用いる。Brandes のアルゴリズムでは Dijkstra 法により経路探索を行い、探索順にノードをスタックに入れ、探索が終了したらスタックからノードを一つずつポップし媒介数を求める。リンクの媒介数は始点と終点のノードの媒介数より求められる。5行目で関数 $ChooseLink$ を呼び出し、キュー Q 内の媒介数が最小のリンク e を取り出し、6行目でグラフ G' からリンク e を取り除いたグラフ G_{tmp} とし、7行目で G_{tmp} を引数に関数 $BetweenCentrality$ を呼び出し、すべての二頂点間の最短経路、最短経路長 $D_{u,v}^{G_{tmp}}$ 、および媒介数を求める。8~10行目では各二頂点間において、7行目で求めた最短経路長 $D_{u,v}^{G_{tmp}}$ と3行目で求めた最短経路長 $D_{u,v}^G$ とを比較し、すべての二頂点間の経路長がグラフ G_{tmp} においてグラフ G から k 以内の増加である場合、9行目でグラフ G' を更新する。 Q が空集合になるまで、5~10行目の操作を繰り返す。 Q が空集合であれば終了し、グラフ G' を出力する。

提案アルゴリズム DeleteLink の計算量を見積もる。アルゴリズム DeleteLink 内では、キューからリンクを一本取り出し、取り出されたリンクに対して $BetweenCentrality$ の呼び出しと最短経路長の増加の確認を行うという操作を、キューが空集合になるまで繰り返す。 $BetweenCentrality$ においては最短経路を探索し、媒介数を計算している。経路探索においては各ノードを始点として全ノードへの経路をダイクストラ法を用いて計算しているため、計算量は $O(|V|^2)$ である。 $BetweenCentrality$ は基本的に経路探索に計算時間を要する。 $BetweenCentrality$ はすべてのノードを始点として経路探索を繰り返し行うので、計算量は

$O(|V|^3)$ である。次に、最短経路長の比較は各二頂点の最短距離の比較を行うので、 $O(|V|^2)$ 時間で実行可能である。4行目からの $while$ ループはリンク集合のキュー Q が空集合になるまで操作を繰り返すので、操作回数は $|E|$ である。ゆえに、提案アルゴリズム DeleteLink の計算量は $O(|E||V|^3)$ である。

3.4 スロット割り当て問題の解法

本節ではスロット割り当て問題を解く提案アルゴリズム AssignSlot を説明する。

先に、スロット割り当て問題における衝突度を定義する。CATBS ではグラフ内の隠れ端末問題の関係にあるリンクペアの数を衝突度とした。スロット割り当て問題においては、リンクの衝突度とノードの衝突度を追加で定義する。リンクの衝突度とは、あるリンクが妨害するリンクの数を表す。ノードの衝突度とは、あるノードを始点として出るリンクの衝突度の総和を表す。

本アルゴリズムの手順を Algorithm 2 に示す。提案アルゴリズムでは、入力にリンク削減問題で得た出力グラフを用いる。このグラフ内のそれぞれのリンクの衝突度を計算し、このリンクの衝突度から各リンクの始点となるノードに足し合わせ、ノードの衝突度を計算する。そして、ノードの衝突度がもっとも高いノードからスロットを決める。ここで、あるノードから出るすべてのリンクのスロットは始点となるノードと同じスロットとする。また、スロットの選び方はノードにそれぞれのスロットを割り当てたときの衝突度を計算し、もっとも衝突度が小さいときのスロットを選ぶ。

提案アルゴリズムを具体的に説明する。1行目で初期化を行い、リンク集合内のリンクをキュー Q に入れる。2行目で関数 $CalcNodeInterference$ を呼び出し、各ノードの衝突度を計算し、3行目で Q を引数に関数 $SelectNode$ を呼び出し、 Q 内のノードでもっともノードの衝突度の高いノード n を返す。5行目で関数 $CalcInterference$ を呼び出す。関数 $CalcInterference$ ではノード n に各スロットを割り当てたときのリンクの衝突度を求めて、リンクの衝突度からノードの衝突度とグラフの衝突度をそれぞれ求める。もっともグラフの衝突度が小さいときのスロット s を戻り値として返し、ノード n のスロットを s とする。6行目でノード n から出るすべてのリンクのスロットを s とする。7行目で Q を引数に関数 $SelectNode$ を呼び出し、もっともノードの衝突度の高いノード n を返す。5~7行目の操作を繰り返し、ノード n が NULL であったら終了する。

提案アルゴリズム AssignSlot の計算量を見積もる。アルゴリズム AssignSlot 内では関数 $CalcInterference$ を呼び出して衝突度を計算し、 $SelectNode$ を呼び出し、スロットを割り当てるという操作を、すべてのリンクにスロットが割り当てられるまで繰り返す。AssignSlot の計算時間は衝突

Algorithm 2: Assign Slot

Input: $G' = (V, E')$ **Output:** $f : E' \rightarrow S$

```
1  $Q \leftarrow V$ 
2 CalcNodeInterference
3  $n \leftarrow \text{SelectNode}(Q)$ 
4 while ( $n \neq \text{NULL}$ )
5    $s \leftarrow \text{CalcInterference}(n)$ 
6    $e \leftarrow s$  ( $e \in$  ノード  $n$  から出るリンク)
7    $n \leftarrow \text{SelectNode}(Q)$ 
8 end while
```

度の計算に依存する．ゆえに，CalcInterference の計算時間を考える．

CalcInterference は，SelectNode で選んだノード n に全スロットをそれぞれ割り当てた時の衝突度を計算し，もっともグラフの衝突度が低かったときのスロットを返す．CalcInterference では，まず，リンクの衝突度を計算する．リンクの衝突度は，あるリンクに対して妨害関係にあるリンク数を求めるので，計算量は $O(|E|^2)$ である．次に，リンクの衝突度を各リンクの始点ノードに足し合わせることでノードの衝突度を求める．ノードの衝突度は，あるノードに対してそのノードから出るリンクの衝突度を足し合わせるので，計算量は $O(|V|)$ である．さらに，リンクの衝突度の総和からグラフの衝突度を求める．グラフの衝突度はリンクの衝突度の総和であるので，計算量は $O(|E|)$ である．CalcInterference はこれらの 3 つの衝突度を，ノード n に各スロットを割り当てたときそれぞれを求めるので，衝突度の計算を $|S|$ 回行う．ゆえに，CalcInterference の計算量は $O(|E|^2|S|)$ である．無線メッシュ網では一般に $|E| = O(|V|)$ であるので，CalcInterference の計算量は $O(|V|^2|S|)$ と表せる．

CalcInterference はすべてのノードのスロットが割り当てられるまで繰り返されるので， $|V|$ 回行う．ゆえに，提案アルゴリズム AssignSlot の計算量は $O(|V|^3|S|)$ となる．

4. 評価

4.1 評価方法

提案手法を評価する．無線メッシュ網は固定基地局によるネットワークであるため，そのトポロジは頻繁には変化しないが，電波状況の変化やノードの故障等の要因により，一時的にトポロジが変化する状況が考えられる．本研究では，このような状況において，各ルータにおいてスケジューリングと経路の再計算を行い，一時的に変化したトポロジにおいても適切な経路を用いて通信できる適応的な経路制御を目指している．これを実現するためには，スケジューリングアルゴリズムを経路制御プロトコルに組み込んで動作させる必要があるが，そのためには，スケジュー

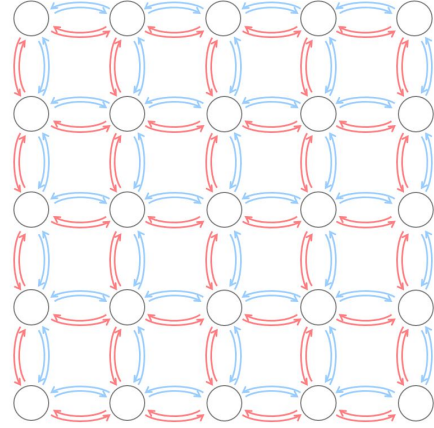


図 4: スロット数が 2 の無線メッシュ網をモデル化したグラフ

リングアルゴリズムは十分に少ないメモリ領域を用いて高速に動作し，それなりに良い解を得る能力を持つことが求められる．本節では，提案アルゴリズムがこのような性質を持つかどうかを評価する．

評価で用いる提案手法は C 言語を用いて実装した．無線メッシュ網を示すネットワークをモデル化したグラフを入力とし，このグラフと，最短経路の許容増加量 k とスロット数を提案手法に与えた．評価トポロジとしては，計画的に配置された無線メッシュ網を想定する 5×5 のグリッド型トポロジを用いた．本トポロジでは縦横に隣接するノードにのみ通信可能であると想定する．最短経路長の許容増加量 k は 0～10 と変化させ，スロット数を 2～6 と変化させた．ただし， k の値は奇数にしても $k-1$ の場合と等価であるため，偶数の場合のみを評価した．入力グラフの例として，スロット数が 2 の無線メッシュ網をモデル化したグラフを図 4 に示す．

衝突度は，提案手法を用いてスケジューリングを行った場合，干渉の原因となる隠れ端末問題の関係にあるリンクペアの数をどれくらい減少させることができたかを測る．このとき，提案手法と，CATBS を用いてスケジューリングを行った場合での衝突度を比較することで提案手法の性能を評価する．また，提案手法を用いてスケジューリングを行う際にかかる計算時間，メモリ消費量を測り，評価する．比較手法は，トポロジを入力して部分 MAX-SAT の入力式を予め作成し，越村氏が作成した部分 MAX-SAT のソルバである qmaxsat[6] を用いてスケジュールを計算した．CATBS の近似解を得るための実行時間が数秒から数十時間と差異があったため，ソルバの実行は 120 秒に制限し，計算がこれを超える場合にはその時点での結果を出力した．

4.2 結果と考察

まず，衝突度について評価する．提案手法を用いて， k の値とスロット数を変化させながら衝突度を求めた．図

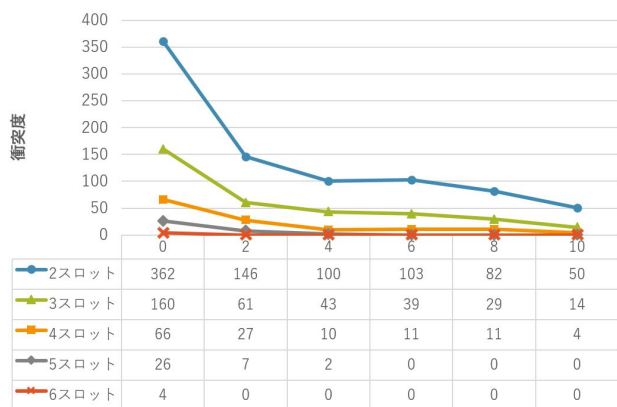


図 5: 提案手法において k を変化させたときの衝突度

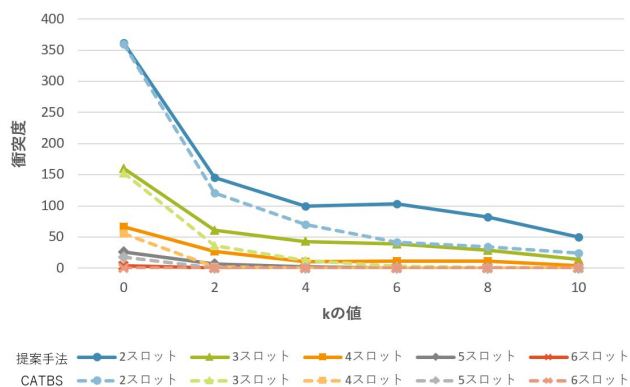


図 6: 2～6 チャンネルでの CATBS との衝突度の比較

5 では、提案手法において各スロットごとに対して k を 0, 2, 4, 6, 8, 10 と変化させたときのそれぞれの衝突度の値を表す。5×5 の入力グラフに対して、 k の値を変化させたとき、5 スロット使用したときに衝突度が 0 になることが確認できる。

また、提案手法と CATBS を用いて、干渉する関係にあるリンクペアを最小化したグラフの衝突度を求めた。図 6 では、提案手法と CATBS において各スロットごとに対して k を 0, 2, 4, 6, 8, 10 と変化させたときのそれぞれの衝突度を比較している。提案手法によって得られた結果を実線で、CATBS によって得られた結果を破線で表している。提案手法が 5 スロット使用し k が 6 以上であるとき衝突度が 0 になったのに対し、CATBS では 4 スロット用いたとき k の値を 4 以上にすると衝突度が 0 になることが確認できる。また、提案手法は CATBS と比べたとき、 k の値に対して衝突度の減少率は低いことが確認できる。これは CATBS が全探索を行う部分 MAX-SAT を用いてスケジューリングを行うことに対して、提案手法は貪欲法を用いており、解候補が局所的な解に陥りやすいことが原因であると考えられる。スケジューリング後のグラフを図 7～図 9 に示す。

次に、計算時間について評価する。スケジューリングに要する時間はルーティングプロトコルにとって重要な要素

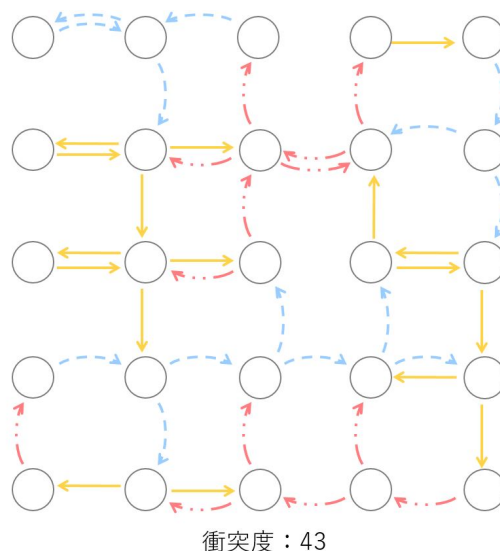


図 7: 提案手法における $k=4, 3$ スロット用いたときの出力グラフ

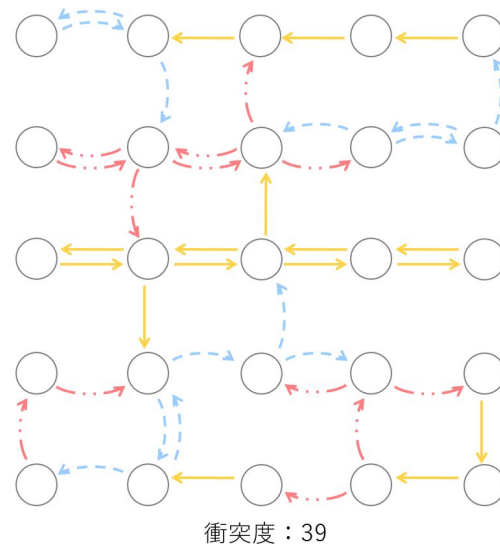


図 8: 提案手法における $k=6, 3$ スロット用いたときの出力グラフ

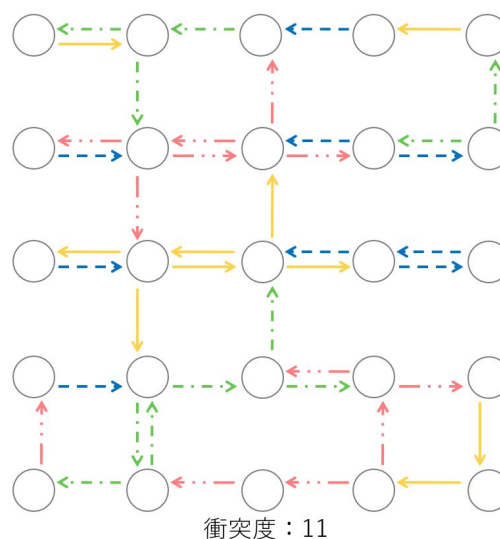


図 9: 提案手法における $k=6, 4$ スロット用いたときの出力グラフ

表 1: 提案手法の計算時間 (秒)

	2slot	3slot	4slot	5slot	6slot
k=0	0.0838	0.081	0.0793	0.0831	0.0779
k=2	0.0856	0.084	0.0843	0.0861	0.0841
k=4	0.0822	0.0848	0.0828	0.0841	0.0835
k=6	0.1014	0.1	0.0968	0.0945	0.097
k=8	0.0879	0.0907	0.0885	0.0891	0.0902
k=10	0.0862	0.0907	0.0902	0.0881	0.0926

の一つである。無線ネットワークはリンクが切断された際に必要に応じて通信路を再計算する。リンクの切断が発生した場合、即座に影響が及ぶ範囲は概ね数ホップであり、経路情報の伝達にかかる時間はおよそ 4~5 秒である。このような伝達に対し、スケジューリングのための計算時間は十分短くなければならない。なぜなら、計算時間が長ければ、伝達にかかる時間は長くなり、古いスケジューリングが適用され続けることで正常に通信を行えないために、通信性能の低下を招くためである。また、CATBS の近似解を得るための実行時間が数秒から数十時間とかかることがあるため、本評価ではルーティングプロトコルに組み込むことを想定するため、ソルバの実行は 120 秒に制限し計算がこれを超える場合にはその時点での結果を出力した。提案手法の計算時間を、表 1 に示す。提案手法と CATBS の計算時間を比較したとき、4 スロット以上になると CATBS の方が短い時間で計算を行う。また、CATBS はスロット数が少ないとき、また k の値が小さい時に計算時間が非常に長いことが確認できる。これに対して提案手法は一定時間で計算できることが確認できる。

さらに、メモリ消費量においては提案手法は CATBS の 1000 分の 1 程度であることが確認できた。

以上から、提案手法は短い計算時間と少ないメモリ消費量で CATBS と近い精度のスケジューリング結果を得られることが確認された。本評価では、提案手法は少ないメモリ消費量で CATBS におけるスケジューリング問題を解くことが出来ることを示した。

5. おわりに

本研究では、CATBS におけるスケジューリング問題を少ないメモリ消費量で解くために、媒介中心性に基づいた貪欲なスケジューリングアルゴリズムを提案した。媒介中心性を用いることにより、経路としてよく使われるリンクを残して通信経路を制限し、残ったリンクに対して隠れ端末問題が発生しないようにスロットを割り当てることでネットワーク内の干渉を削減する。これを実現するために、本研究では CATBS におけるスケジューリング問題を二つに分割し、それぞれの問題に対して貪欲アルゴリズムを与えて解いた。評価の結果、比較的にかさい無線メッシュ網を想定した 5×5 の格子状のグラフにおいては、5 スロッ

ト用いた際に衝突度が 0 になることを確認した。また、計算時間を計測し、ルーティングプロトコルに組み込むスケジューリング法として有効である可能性を示した。

しかし、本研究は CATBS との比較から、衝突度において削減の余地が残されていると考えられる。本研究では、リンクの削減において媒介中心性という指標を用いて行ったが、スロットの割り当てにおいては精度を上げる余地があると考えられる。これは今後の課題としたい。また、提案アルゴリズムはリンクを削減する際において、ダイクストラ法を用いた経路探索を何度も行っているため、4.2 節で計算したように計算量が $O(|E||V|^3)$ にかかる。さらに削減できる余地が残されていると考え、今後の課題としたい。

謝辞 本研究の一部は JSPS 科研費 16K12422 の助成によるものである。ここに記して謝意を表す。

参考文献

- [1] I. F. Akyildiz and X. Wang, "Wireless Mesh Networks," John Wiley & Sons, Ltd, Publication, 2009.
- [2] IEEE802.11 Wireless Local Area Networks, 入手先 (<http://www.ieee802.org/11/>) 2017-2-15 参照.
- [3] Takuya Yoshihiro and Taiki Nishimae.: *Practical Fast Scheduling and Routing over Slotted CSMA for Wireless Mesh Networks*, IEEE/ACM International Symposium on Quality of Services (IWQoS2016) (2016).
- [4] Freeman, L.C.: *Social Networks*, Vol. 1, pp.215-239, (1979).
- [5] Ulrik Brandes: *A Faster Algorithm for Betweenness Centrality*, Journal of Mathematical Sociology., pp163-177, (Dec 2000).
- [6] Yukihide Koshimura: *QMaxSAT: A Partial Max-SAT Solver*, Journal on Satisfiability, Boolean Modeling and Computation, Vol.8, pp.95100, (2012).