

伝送周期の調整による無線マルチホップネットワーク上での TCP の輻輳の緩和手法

中原 史博¹ 北須賀 輝明¹ 有次 正義¹

概要：無線マルチホップネットワークにおいて、TCP プロトコルによる双方向通信では、ホップ数の増加に伴いスループットが低下する。本研究では、単一および複数の TCP ストリームを対象に、輻輳を緩和することを目的とし、各ノードのデータリンク層で次のノードへのパケット伝送までに待ち時間を入れる手法を提案する。シングルストリームとマルチストリーム上で待ち時間を変化させて、スループットとパケット伝送数をシミュレーションし、その結果を元に待ち時間を設定し、提案手法による輻輳の緩和とスループットについて評価した。結果から、単一および複数の TCP ストリームのすべてのシミュレーションで、スループットは少し減少したものの、パケット伝送を減らし、輻輳を緩和させることが示された。

A Method of Alleviating Congestion of TCP Stream on Wireless Multi Hop Networks by Adjusting the Transmission Interval

FUMIHIRO NAKAHARA¹ TERUAKI KITASUKA¹ MASAYOSHI ARITSUGI¹

1. はじめに

無線マルチホップネットワークとは、送信元ノードと宛先ノードが通信する際に他の端末を経由して通信する無線ネットワークである。このようなネットワークでは、中継ノードを経由してパケットを伝送するため、広範囲の通信が可能となるだけでなく、基地局を必要としないため低コストでネットワークを構築することができる。また、災害等で中継ノードに故障が生じた場合、別の中継ノードを用いることでネットワークの再構築が容易であるという利点がある。

一般に無線マルチホップネットワークでは、ホップ数が増えるにつれスループットが低下する。スループット低下の原因として、二つのノードを結ぶ中継経路上での電波干渉（フロー内干渉）や、異なる経路間での電波干渉（フロー間干渉）による輻輳が生じることがあげられる。また、TCP プロトコルを用いた通信では送信元ノードから宛先ノードに向けたパケット伝送と、宛先ノードから送信元ノードにむけた肯定応答からなる双方向通信のため、送信元ノード

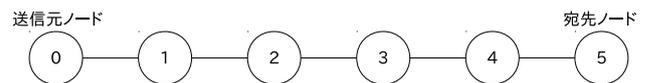


図 1 線トポロジー

から宛先ノードに向けてのみ伝送する UDP プロトコルに比べて、フロー内干渉による衝突が頻繁に起きる。

このような輻輳に対して、古川 [1] や比嘉ら [2], 丸田ら [3] は送信元ノードがバッファに蓄えたパケットを間を空けて周期的に送信し、周波数リユース間隔を制御することで、各中継ノード間のフロー内干渉量を制御する周期的間欠送信法 (Intermittent Periodic Transmit) を提案している。また Rangwala ら [4] は、メッシュ状に形成された無線マルチホップメッシュネットワークにおいて、近傍の輻輳を感知し、輻輳情報を共有する伝送レート制御プロトコル Wireless Control Protocol (WCP) と、トポロジーの形状から近傍の利用可能な通信容量を推定し、伝送レートを公平に配分する Wireless Control Protocol with Capacity estimation (WCPCap) により輻輳を緩和する手法を提案

¹ 熊本大学大学院自然科学研究科
Graduate School of Science and Technology, Kumamoto University

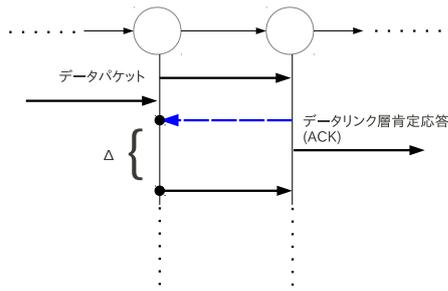


図 2 提案手法によるパケット送信の調節方法：各ノードのデータリンク層に ACK を受信する毎に次のパケット送信を待ち時間 Δ 遅らせる

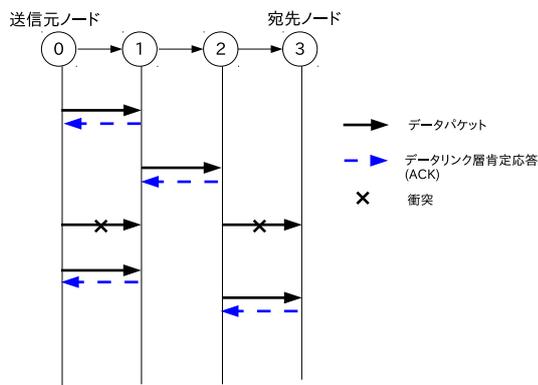


図 3 フロー内干渉によるパケット衝突と再送の様子

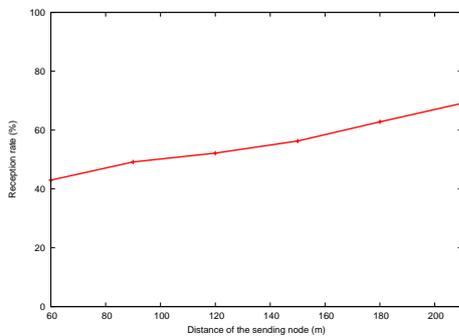


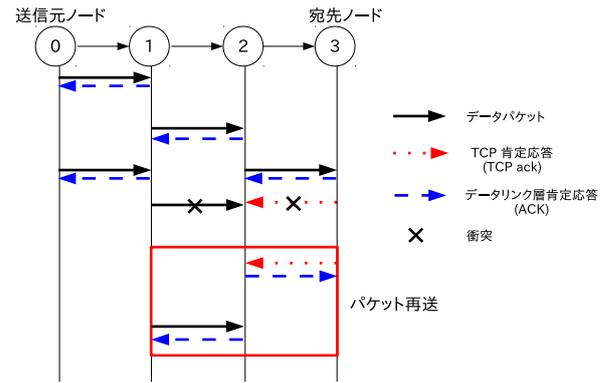
図 4 送信ノード間距離に対するパケット受信率

している。

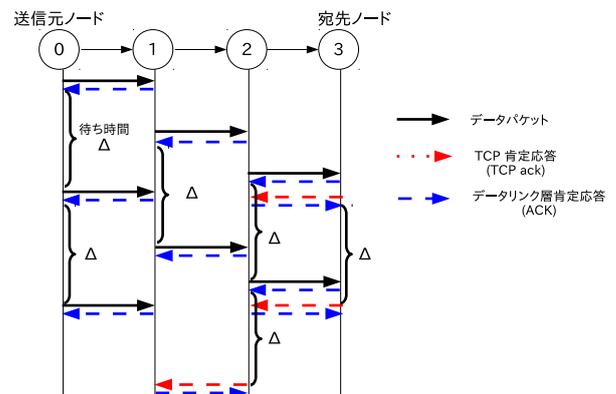
本研究では、シングル TCP ストリームとマルチ TCP ストリームを対象に、フロー内干渉を減らし、輻輳を緩和することを目的とする。本研究ではデータリンク層で送信間隔を空ける手法を提案し、パケット伝送数の比較による輻輳の緩和とスループットをシミュレーションにより評価する。

2. 提案手法

この章では、無線マルチホップネットワーク上での TCP プロトコルの通信によって生じる輻輳を緩和させる手法を提案する。



(a) データパケットと TCP ack の衝突によるパケット再送の様子



(b) 提案手法では待ち時間 Δ を入れることによりデータパケットと TCP ack のパケット衝突を減らす

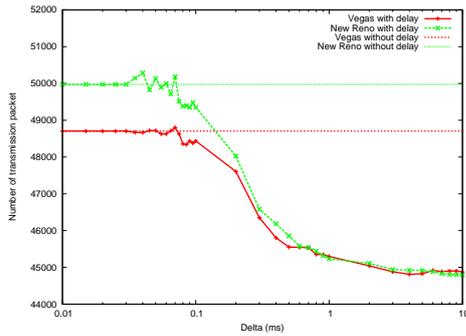
図 5 データパケットと TCP ack の衝突

2.1 パケット送信時間の調整

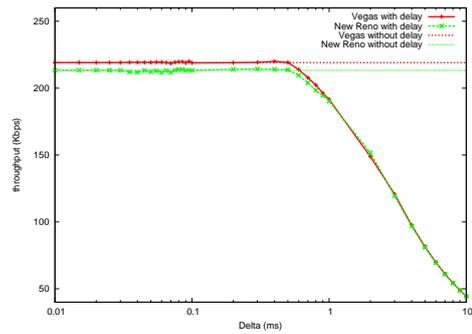
TCP プロトコルにおいてデータの受信が正しく行われた際、宛先のノードは送信元のノードに向けて肯定応答 (TCP ack) を送信する。本研究では、図 1 のような線トポロジーのノード配置を考える。ノード 2 からノード 3 のように隣のノードに向けたパケット伝送に対するデータリンク層の肯定応答を ACK と表し、宛先ノードであるノード 5 から送信元ノードであるノード 0 への TCP プロトコルの肯定応答を TCP ack と表記する。

提案手法では、図 2 のように各ノードのデータリンク層で ACK を受信してから、次のパケット伝送までに、 Δ 秒間待つ。この待ち時間によりデータリンク層で送信間隔が空き、フロー内干渉を抑制できると考えられる。このようにして TCP プロトコルにおける輻輳を緩和させる。

フロー内干渉は図 3 のような状態を指す。図 1 のようなトポロジーでノード番号が端から 0 から 8 までであり、近傍ノード間距離が 30 m であるトポロジーを例として 0 番から 1 番までの通信と n 番から $n+1$ 番の通信を同時にする場合を考える。送信ノード間距離 (0 番と n 番の距離) は $n \times 3$ m となり、図 4 に 10 秒間 UDP によるパケット伝送をした場合の、送信ノード間距離におけるパケット受信



(a) パケット伝送数



(b) スループット

図 6 2 ホップのシングルストリームにおける待ち時間 Δ によるスループットとパケット伝送数

率を示す. このように送信ノード間の距離が近いほど, フロー内干渉によるパケットの衝突が生じる.

パケットの伝送の周期を空け, フロー内干渉を抑えるという点では, 文献 [1], [2], [3] と似ているが, これらは送信元ノードでパケット送信の周期を空けているのに対して, 提案手法では送信元ノードだけではなく各中継ノードでもパケットを伝送する間隔を空ける. このため, 双方向通信である TCP プロトコルで生じる図 5(a) のような, データパケットと TCP ack の衝突と, それに伴う再送を, 図 5(b) のように減らせることが期待される.

2.2 待ち時間の決定について

次に待ち時間 Δ の設定方法について述べる. 提案手法を用いて, 本論文の目的である輻輳の緩和を実現するためには, 待ち時間 Δ の設定が重要となる. 本研究では待ち時間 Δ は次で述べるシミュレーションにより定める.

表 1 のシミュレーション環境において, 図 1 のような線トポロジーで, 近傍ノード間距離を 30 m, ノード数 3 とし, 真ん中のノードを中継して両端ノード間で通信する場合を考える. このような 2 ホップのシングルストリーム上において, Δ を $10 \mu\text{s}$ から $95 \mu\text{s}$ まで $5 \mu\text{s}$ 刻み, $100 \mu\text{s}$ から $900 \mu\text{s}$ まで $100 \mu\text{s}$ 刻み, そして 1 ms から 10 ms まで 1 ms 刻みで変化させた場合のスループットとパケット伝送数を図 6 に示す. なお, パケット伝送数は, 各ノードで伝送したデータパケット数と TCP ack 数の合計である. また, 待ち時間を設定しない従来手法の値を横の直線で示している.

図 6(a) のように, Δ が大きくなれば, フロー内干渉を抑え, パケット伝送を減らすことができるため, 輻輳を緩和することができる. しかし, 図 6(b) のように Δ が大きすぎると, スループットが低下する. これは, データリンク層で ACK を受信してから, 次のパケット伝送までに遅延時間が発生するためである. 一方で, Δ が小さすぎる場合は, 輻輳を緩和することはできない. また, TCP New Reno, TCP Vegas では, パケット伝送数やスループットに違いはある

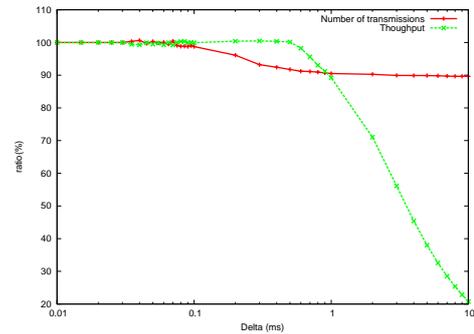


図 7 従来手法に対するパケット伝送数とスループットの割合

表 1 シングルストリームでのシミュレーション環境

通信規格	IEEE 802.11
ルーティング	静的ルーティング (NOAH)
TCP	New Reno, Vegas
データサイズ	10 M バイト
近傍ノード間距離	30 m

が, Δ の変化に伴う値の変化は, 似た傾向を示した.

New Reno について, 従来手法に対するパケット伝送数とスループットのそれぞれの割合を図 7 に示す. $100 \mu\text{s}$ から $600 \mu\text{s}$ の間は, スループットは従来手法とほぼ同じ値だが, 約 2,000 (4%) から約 4,000 (10%) のパケット伝送数を削減できている. このように, Δ の増加に伴い, パケット伝送数が低下し始めるのが, スループットに比べ早いことに注目し, パケット伝送数が減少し始め, スループットが従来手法に比べてあまり減少しない Δ の範囲を定める. Δ の増加によるパケット伝送数の減少は, 単調に減少する. 従って, パケット伝送数が 100%未満でかつ, Δ の増加に伴い, 連続して 3 回 パケット伝送数を減らし始める点を Δ_a とし, Δ_a から始まり, スループットを 95%未満にする Δ の値の 1 つ前を Δ_b とし, 次式で Δ_x を求める.

$$\Delta_x = \frac{\Delta_a + \Delta_b}{2} \quad (1)$$

式 (1) で求まる Δ_x を本手法の待ち時間として用いる. 次章で, 単一および複数の TCP ストリームを対象にシミュレーションし, Δ の変化に伴うパケット伝送数とスループット

表 2 マルチストリームでのシミュレーション環境

通信規格	IEEE 802.11
ルーティング	静的ルーティング (NOAH)
TCP	Vegas
データサイズ	フロー A : 4 M バイト フロー B : 5 M バイト フロー C : 3 M バイト
ホップ数	フロー A : 5 ホップ フロー B : 5 ホップ フロー C : 5 ホップ
近傍ノード間距離	30 m

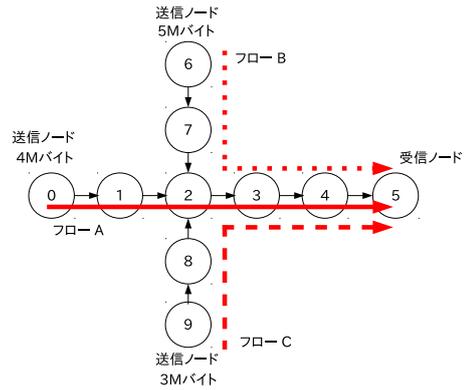


図 8 マルチストリームのシミュレーションのノード配置

表 3 シミュレーションより算出した Δ_a , Δ_b と Δ_x

		Δ_a	Δ_b	Δ_x
シングルストリーム (New Reno)	2 ホップ	95 μ s	600 μ s	347.5 μ s
	5 ホップ	2 ms	4 ms	3 ms
	9 ホップ	2 ms	5 ms	3.5 ms
	14 ホップ	2 ms	7 ms	4.5 ms
シングルストリーム (Vegas)	2 ホップ	100 μ s	800 μ s	450 μ s
	5 ホップ	900 μ s	3 ms	1.95 ms
	9 ホップ	1 ms	3 ms	2 ms
	14 ホップ	2 ms	6 ms	4 ms
マルチストリーム	全フロー	1 ms	4 ms	2.5 ms

ットから Δ_x を求め、提案手法によるパケット伝送数とスループットを評価する。

3. シミュレーション

この節では、単一および複数の TCP ストリームを対象に、待ち時間 Δ の変化によるスループットとパケット伝送数を求めた後 Δ_x を算出し、提案手法によるスループットとパケット伝送数を評価する。

3.1 シミュレーション条件

シミュレーションには、ネットワークシミュレータ ns-2(ver. 2.34) を用いて通信規格 IEEE 802.11 を対象にシミュレーションする。IEEE802.11 のシミュレーションは Chen ら [5] の実装を用いる。提案手法による輻輳緩和について評価するため、輻輳制御には、ルーティングパケットを使用せず、静的にルーティングする NOAH (NO Ad-Hoc Routing Agent)[6] を用いた。待ち時間 Δ は先ほどと同じく、10 μ s から 95 μ s まで 5 μ s 刻み、100 μ s から 900 μ s まで 100 μ s 刻み、1 ms から 10 ms まで 1 ms 刻みで変化させる。シングルストリームについては先ほどと同じ、表 1 に示す条件でシミュレーションする。ホップ数は 2, 5, 9, 14 の 4 通りとする。マルチストリームについては表 2 で示す条件でシミュレーションする。マルチストリームのシミュレーションは、図 7 に示す近傍ノード間距離 30 m のノード配置で、通信開始後 1 秒後にノード 0 からノード 5 に向けた通信 (フロー A) を開始し、その 5 秒後に、ノード 6 からノード 5 に向けて通信 (フロー B) を開始し、フロー B の 4 秒後にノード 9 からノード 5 への通信 (フロー C) を開始する。待ち時間 Δ はシングルストリームの場合と同様に变化させ、従来手法と比較したパケット伝送数とスループットの改善率を各フローごとと、フロー全体で評価する。マルチストリームでは、TCP Vegas のみ扱う。

3.2 待ち時間 Δ の変化に伴うパケット伝送数とスループットの関係

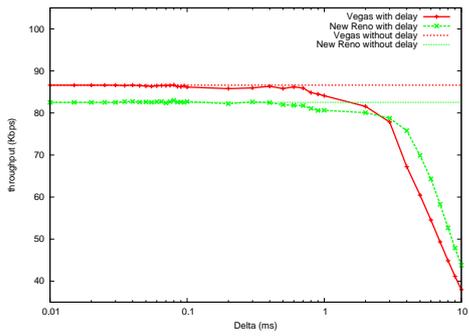
Δ の変化に伴うパケット伝送数とスループットの関係についてのシミュレーション結果を示す。シングルストリー

ムの 5 ホップから 14 ホップまでの結果を、図 3.2 から図 9 に、マルチストリームにおける結果を、図 10 に示す。スループットは従来手法と比べてほとんど向上せず、例えば、TCP Vegas の 10 ホップの場合の Δ が 95 μ s の場合において、従来手法と比べ、0.3 % のスループットを向上させたが、これは、約 210 秒の通信時間に対して、0.6 秒程度の通信時間短縮しか得られず、効果的な結果は得られなかった。パケット伝送数については、TCP Vegas の 14 ホップの場合の Δ が 300 μ s で従来手法のパケット伝送数、約 350,000 に対して、約 9,000 (3 %) のパケット伝送を削減し、スループットが従来手法と同等に保たれていることから、従来手法よりも、輻輳を緩和することができている。また、5, 9, 14 ホップのパケット伝送数は、70 μ s で減少し始め、7 ms で減少がゆるやかになる 2 ホップの場合とは減少傾向は異なるが、 Δ の増加に対して、スループットに比べて早く減少し始めている点は同じである。

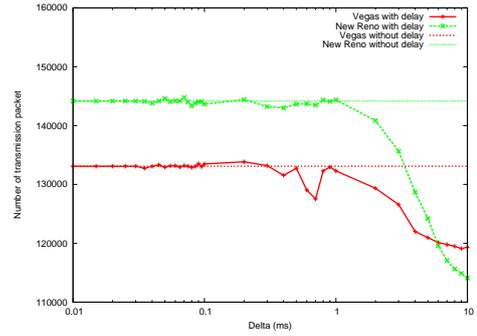
3.3 本手法によるシミュレーション

先のシミュレーション結果から算出した、 Δ_a , Δ_b 及び、本手法の待ち時間 Δ_x を表 3 に示す。

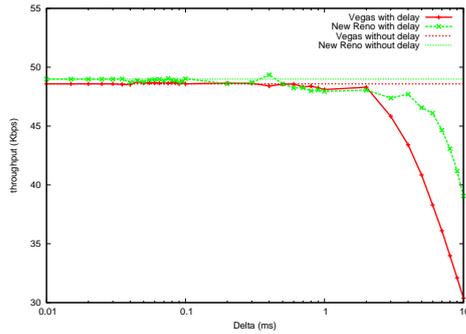
なお、マルチストリームについては、各フロー毎ではなく、フロー全体についてのものを算出している。シン



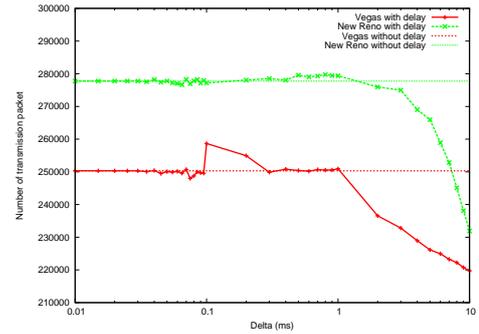
(a) 5 ホップのスループット



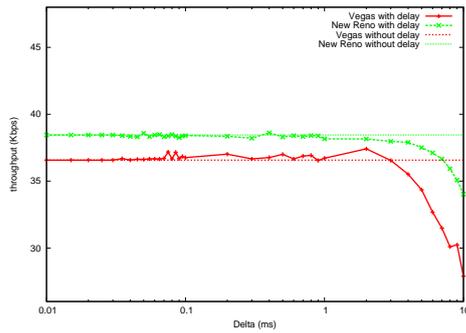
(b) 5 ホップのパケット伝送数



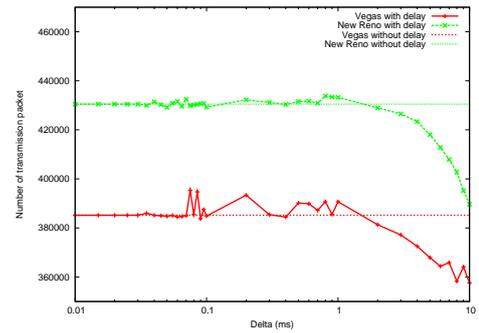
(c) 9 ホップのスループット



(d) 9 ホップのパケット伝送数

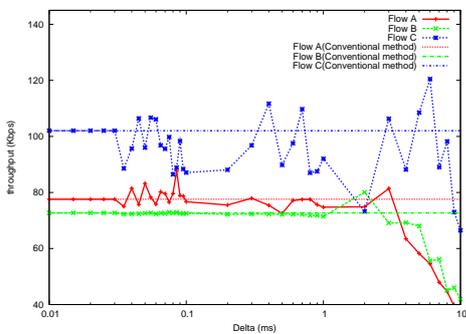


(e) 14 ホップのスループット

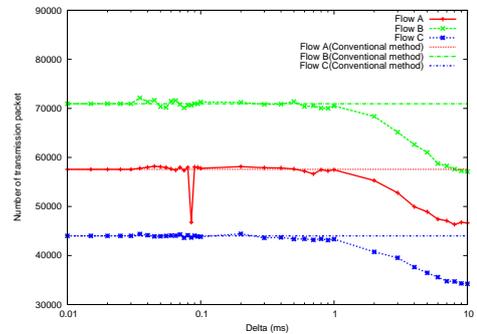


(f) 14 ホップのパケット伝送数

図 9 14 ホップのシングルストリームにおける待ち時間 Δ によるスループットとパケット伝送数

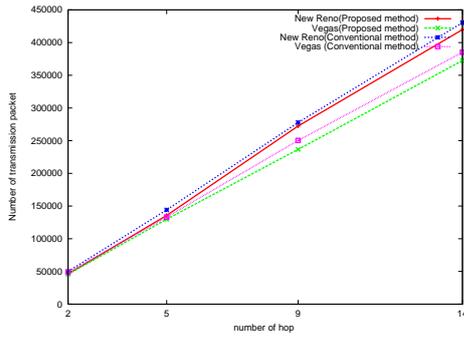


(a) スループット

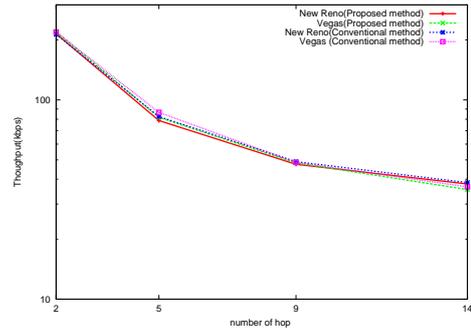


(b) パケット伝送数

図 10 マルチストリームにおける待ち時間 Δ によるスループットとパケット伝送数

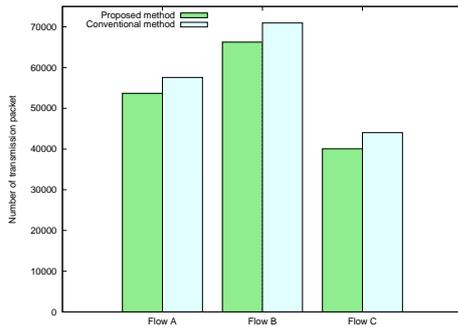


(a) パケット伝送数

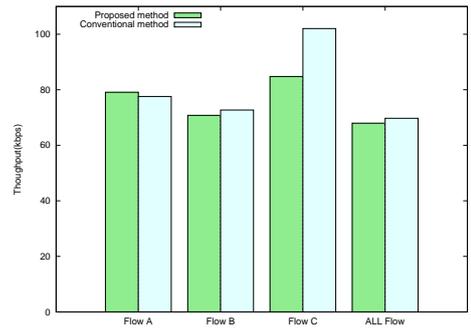


(b) スループット

図 11 シングルストリームにおける提案手法と従来手法の比較結果



(a) パケット伝送数



(b) スループット

図 12 マルチストリームにおける提案手法と従来手法の比較結果

グルストリームにおいては、ホップ数が増えるにつれ、待ち時間 Δ_x は増加している。また、マルチストリームについては、各フローは 5 ホップであるが、シングルストリームの TCP Vegas の 5 ホップの場合と比べて、 Δ_x は長く、5 ホップの場合より、フロー内干渉が生じやすい 10 ホップの場合に近い値を示した。

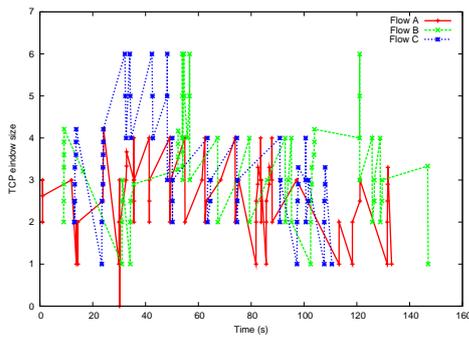
Δ_x を本手法の待ち時間とし、提案手法により待ち時間を設定しない従来手法と比べ、輻輳の緩和が実現しているか評価する。シングルストリーム (New Reno) の 2, 9, 14 ホップとシングルストリーム (Vegas) の 2, 5 ホップ、さらに、マルチストリームの場合については、新たにシミュレーションし、パケット伝送数による輻輳の緩和の評価と、スループットを評価する。

シングルストリームにおける提案手法と従来手法のパケット伝送数とスループットを図 11 に示す。TCP New Reno, TCP Vegas 共にすべてのホップ数で、パケット伝送数を減らすことができた、スループットについては、各輻輳制御の 2 ホップの場合は提案手法は従来手法よりスループットが向上したが、他の場合では若干下回る結果となった。次にマルチホップについてのシミュレーション結果を図 12 に示す。マルチストリームでは、各フローでパケット伝送数を減らすことができたが、スループットの向上には繋がらず、フロー C については大きく低下する結果となった。

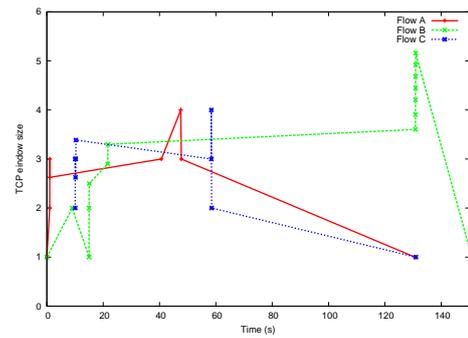
3.4 考察

表 4 に提案手法が従来手法より減らすことができたパケット伝送数と、従来手法と比べてどのくらいのスループットを得ているかを示す。すべての結果においてパケット伝送数を削減することができ、輻輳の緩和という目的は達成できた。シングルストリームについて、2 ホップの場合は TCP New Reno と TCP Vegas のどちらでも、スループットを向上させているが、他のホップ数の場合では、スループットは増加しなかった。ホップ数が増加するほど、パケットの伝送数は増加し、例えば、TCP New Reno の場合、ホップ数 14 のときは、ホップ数 2 に比べ、10 倍近いパケットを伝送する。ところが、New Reno の 14 ホップ時に削減できた伝送数は 2 ホップの場合の削減数の 3 倍にも満たない。一方、TCP Vegas の 10 ホップの場合は、パケットの伝送数は、2 ホップ時の 5 倍程度なのに対して、伝送の削減数は、10 倍以上ある。このように提案手法により輻輳が緩和できても、ホップ数が多ければ、データリンク層で ACK を受信してからパケットを伝送するまでに、待ち時間が発生する場面が多くなるため、スループットの向上も目指すには、待ち時間 Δ の動的な制御が必要であると考えられる。

次にマルチストリームについて考察する。今回マルチストリームの全フローで、従来手法が 170,000 のパケットを伝送したのに対して、提案手法では 12,500 程度の伝送を削減でき、各フローを見ても、パケット伝送数が増加した



(a) 従来手法



(b) 提案手法

図 13 従来手法と提案手法の時間変化による TCP ウィンドウサイズの変化

表 4 従来手法に対するパケット伝送数とスループット

		パケット削減数	スループット
シングルストリーム (New Reno)	2 ホップ	92.3 %	100.8 %
	5 ホップ	94.1 %	95.5 %
	9 ホップ	98.1%	97.0 %
	14 ホップ	97.1 %	98.5 %
シングルストリーム (Vegas)	2 ホップ	94.0 %	100.2 %
	5 ホップ	97.6 %	94.5 %
	9 ホップ	94.5 %	99.4 %
	14 ホップ	96.7 %	97.1 %
マルチストリーム	フロー A	93.2 %	101.9 %
	フロー B	93.4 %	97.3 %
	フロー C	91.0 %	83.1 %
	全フロー	92.7 %	97.5 %

フローは存在しない。しかし、フロー C のスループットが 83.1 % に減少し、全体でも低下している。各フローの時間毎の動きを見るため、従来手法と提案手法の時間に対する TCP ウィンドウサイズを図 13 に示す。マルチストリームのシミュレーションでは、すべてのフローの輻輳制御に TCP Vegas を用いている。TCP Vegas はパケットが返ってくるまでの往復時間である Round trip time(RTT) によってウィンドウサイズを変化させる。図 13(a) では最初の通信 フロー A の開始後からフロー B, C が開始されるまでの開始約 10 s と フロー A, C が完了し、フロー B のみが通信する、130 s から 145 s の間を除き、2 つ以上のフローが重なる時間帯ではウィンドウサイズの変化が頻繁に起きている。一方、ACK を受信してから、次のパケット伝送までに、2.5 ms の待ち時間を設定した提案手法では、3 つのフローが重なり合う状況でもウィンドウサイズの変化はあまり見られなかった。従来手法で多く発生したウィンドウサイズの変化が提案手法でほとんどなくなり、パケット伝送数も減らすことができている点から、すべてのフローの輻輳状態の緩和を達成することができたといえる。従って、提案手法で、スループットが低下した理由として、待ち時間による遅延時間の発生だけでなく、TCP Vegas によりウィンドウサイズが変化していないことも原因と考えるこ

とができる。今後は、TCP New Reno の場合や、マルチストリームの多様な通信状況についても検証する必要がある。また、輻輳状態が緩和できたものの、提案手法では全体の通信時間が従来手法に比べ 4 s 程長い約 150 s となったことを踏まえ、スループット向上についても考える必要がある。

4. おわりに

本研究では、無線マルチホップネットワーク上での TCP の輻輳を緩和することを目的とした。その手法として各ノードにおいてデータリンク層の ACK を受信してから次のパケットを伝送するまでに、待ち時間を空けて伝送する手法を提案した。

まず、シングルストリームとマルチストリーム上で、待ち時間による、スループットとパケット伝送数の関係をシミュレーションにより調査した。その結果からパケット伝送数を減らしつつ、スループットを維持するような待ち時間を設定し、シミュレーションによって、スループット向上と輻輳の緩和について評価した。結果としてシングルストリームでは、シミュレーションしたすべての場合でパケット伝送数を減らし、輻輳を緩和することができた。マルチストリームもすべてのフローで輻輳が緩和できた。

今後の課題として、2 ホップのシングルストリームでしか向上できなかったスループットを、他の状態でも向上させるため、待ち時間の動的な制御と、マルチストリームにおける、TCP New Reno を用いた場合の比較や、通信する環境を変化させた場合での検証が挙げられる。

参考文献

- [1] 古川 浩, 「無線マルチホップネットワークにおけるパケット送信間隔制御法の提案」, 電子情報通信学会技術研究報告書, vol.103, no.458, pp.97-102, 2003.
- [2] 比嘉 征規, 古川 浩, 「線トポロジー型無線マルチホップネットワークにおける高効率中継伝送方式」, 電子情報通信学会論文誌, vol.J90-B, no.12, pp.1225-1238, 2007.
- [3] 丸田 一輝, 東坂 悠司, 比嘉 征規, 古川 浩, 「TCP トラフィックを想定した経路予約型周期的間欠送信」, 電子情報通信学会技術研究報告書, vol.106, no.478, pp.111-116, 2007.

- [4] Sumit Rangwala, Apoorva Jindal, Ki-Young Jang, Konstantinos Psounis, and Ramesh Govindan, “Understanding Congestion Control in Multi-hop Wireless Mesh Networks”, Proceedings of the 14th ACM international conference on Mobile computing and networking, pp.291–302, 2008.
- [5] Qi Chen, Felix Schmidt-Eisenlohr, Daniel Jiang, Marc Torrent-Moreno, Luca Delgrossi, and Hannes Hartenstein, “Overhaul of IEEE 802.11 Modeling and Simulation in NS-2”, Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM '07), pp.159–168, 2007.
- [6] No Ad-Hoc Routing Agent (NOAH), <http://icapeople.epfl.ch/widmer/uwb/ns-2/noah>
- [7] Lawrence S. Brakmo, Sean W. O'Malley, Larry L. Peterson, “TCP Vegas: New Techniques for Congestion Detection and Avoidance”, Proceedings of the conference on Communications architectures protocols and applications (SIGCOMM '94) , pp.24–35, 1994.