

救助指揮システムを基にした MANET における遅延極小化 Delay Reduction in MANET Reflecting Chain of Rescue Command

奥田 裕樹†
Yuuki Okuda

原田 史子‡
Fumiko Harada

島川 博光‡
Hiromitsu Shimakawa

1. はじめに

大規模な災害が発生すると、消防隊員は大人数で救助活動をする。大人数の救助活動を指揮する者は、現場の状況を多く知る必要があるが、既存の通信インフラの利用は災害により期待できない。そこで、各隊員が通信端末を持ち Mobile Ad-hoc Networks(以下、MANET)を構築することで現場状況が伝達できる状況を想定する。このとき、伝達されるデータは指揮に用いられるため、遅延欠損があってはならない。本論文では、ネットワークを流れるデータに優先度を設けることで、指揮を執るために必要である重要なデータの遅延欠損がないことを一定の通信量まで保障する方法を提案する。

2. MANET を用いた災害救助

災害救助をする消防隊員の指揮系統は木構造をとり、救助活動は複数の隊員からなる小隊ごとに行われる。複数の小隊は1つの中隊に属し、複数の中隊は1つの大隊に属する。各小隊・中隊・大隊には各隊を指揮する一人の隊長が存在し、指揮はある隊の隊長から下の隊の隊長・隊員に対して執られる。

災害現場では、既存の通信ネットワークが利用できないことが多い。そのため、各隊員が無線通信機能付きの小型計算機を持つことにより、その場で MANET を構築し、さまざまなデータをやりとりすることで、消防活動を支援する研究がある [1]。これらのデータは指揮を執る際に利用されるため、できる限り遅延しないことが求められる。

MANET におけるデータの遅延欠損は、転送ノードの送信キューが満杯になることで発生する。TCP では、転送ノードのキューが満杯になると輻輳制御を行い、データ遅延が発生する。そのため、転送ノードのキューが満杯にならない通信量が、TCP の遅延なく伝達可能なデータ量となる。MANET 環境においてノードが直線的に配置された場合の、TCP スループットは理論的に解析されている [2] が、ネットワークポロジが複雑な場合の遅延なく伝達可能な TCP の最大データ量はわからない。また、UDP は転送ノードのキューが満杯になってもデータ送信をやめないため、データ遅延は発生しないがデータ欠損が発生する。

3. 優先度設定による遅延極小化

3.1 伝達されるデータ

各隊員のデータは、指揮系統に沿って隊員が属する隊の隊長まで、木構造の葉から根の方向へ伝達される。そのため、複数の隊員データが集まり伝達のボトルネックとなるノードができる。

表 1: パラメータ定義

P_x	ノード x のデータ発生量 (bps)
T_x	ノード x の TCP データ発生量 (bps)
S_x	ノード x の送信データ量 (bps)
$P_{max}(x)$	ノード x の送信可能最大データ量 (bps)
$T_{max}(x)$	ノード x の送信可能最大 TCP データ量 (bps)
W	最大通信帯域 (bps)
a	帯域利用効率 ($0 \leq a \leq 1$)
b	TCP 通信の帯域利用効率 ($0 \leq b \leq 1$)
c_x	ノード x の競合ノード数 ($0 \leq c_x$)
$C(x, n)$	ノード x が競合する n 番目のノード ($0 \leq n \leq c_x$)
h_x	ノード x が送信するパケットのホップ数 ($0 \leq h_x$)
$H(x, n)$	ノード x が送信するパケットの n ホップ目ノード ($0 \leq n \leq h_x$)

各隊員が伝達するデータは重要データと一般データの2種類に区別できる。重要データは遅延欠損が許されず、一般データは多少の欠損は許容できる。そこで、重要データには着信保障ができる TCP を、一般データには UDP を用いる。

各隊長は指揮を執るために、できるだけ多くのデータが欲しい。そのため、各隊員からできる限りの一般データが送信され、MANET には大量の UDP データが流れる。そのような環境下では、ボトルネックノードの送信キューが一杯になり輻輳制御により TCP はデータを送信できない。本論文では、データに優先度を設け、送信キューを優先度ごとに分割することで TCP データを最優先で送信する手法を提案し、遅延なく伝達可能な TCP の最大データ量を求める。

3.2 重要データの優先

各隊員が送信するデータには、送信先の隊長が求める度合いにより重要度が存在する。本手法では、データにアプリケーション層で優先度を設けることで、トランスポート層で TCP と UDP を使い分け、重要データの遅延欠損をなくす。設定する優先度には3段階あり、MANET の経路情報をやりとりする HELLO メッセージなどに最高の優先度を設ける。次に TCP を用いる重要データの優先度を設ける。最後に UDP を用いる一般データ用の優先度を設ける。

各ノードの送信キューを優先度の数に分割する。送信・転送するデータは優先度ごとにそれぞれのキューに入る。データは最も優先度の高いキューから順に送信し、優先

$$\forall k(0 \leq k \leq h_x); T_x \leq (Wa - \sum_{n=1}^{c_x} S_{C(H(x,k),n)}) \times b \rightarrow T_x \leq T_{max}(x) \quad (1)$$

$$P_{max}(x) = Wa - \sum_{n=1}^{c_y} S_{C(y,n)} \quad (2)$$

$$S_{C(y,n)} = \begin{cases} P_{max}(C(y,n)) & (\frac{P_{C(y,n)}}{P_{max}(C(y,n))} \geq 1) \\ P_{C(y,n)} & (\frac{P_{C(y,n)}}{P_{max}(C(y,n))} < 1) \end{cases} \quad (3)$$

† 立命館大学大学院 理工学研究所

‡ 立命館大学 情報理工学部

度が高いキューの中身が空にならないと優先度の低いキューの中身を送信しないことで、TCP データを UDP データより優先する。

3.3 遅延しない最大送信データ量

本手法を用いれば TCP データは優先されるが、送信する TCP データ量が多ければボトルネックノードのキューが満杯になり遅延が発生する。そこで、本手法の遅延なしで送信できる最大の TCP データ量を求める。

表1のパラメータを定義すると、ノード x が T_x の TCP データを送信したいとき、 $T_x \leq T_{max}(x)$ ならば T_x は遅延なしで伝達可能である(式(1))。また、ボトルネックノードが y ならば、ノード x が送信可能な最大データ量は式(2)となる。このとき、ノード y が競合するノード n の送信データ量 $S_{C(y,n)}$ は式(3)となる。

4. 最大送信データ量の確認

4.1 想定環境

各隊長はできる限り多くのデータを欲するため、UDP で常に大量のデータが流れている状況を想定する。このような最悪状況下で遅延なく伝達可能な TCP データの最大送信量 T_{max} は、いかなる状況下でも保障される伝達可能なデータ量である。

n 人の隊員がノード x を経由し、ノード y にデータを伝達するシミュレーションを ns2 ver2.34 で行った。ネットワークポロジは図1であり、ノード x がボトルネックノードとなる。通信帯域 W は 2(Mbps) であり、ノード 0~7 とノード x は通信帯域以上である 2(Mbps) のデータを送信しようとする(表2)。ルーティングプロトコルは OLSR であり、TCP・UDP データは経路情報がやりとりされた後である 10(s) から 90(s) 間発生させた。また、TCP データの発生量 T_0 と UDP データを発生させるノードを 1~7 個間に変化させた。そして、ノード y へのデータ到着時間を調べ、すべての TCP データが最後の UDP データよりも早く届いていれば、 $T_x \leq T_{max}(x)$ と考え、 T_x の値を 1(kbyte/s) づつ変化させ、 $T_{max}(x)$ を求めた。

4.2 予想される最大送信データ量

想定環境下において、ノード 0 の送信可能データ量 $T_{max}(0)$ は $P_{max}(0)$ と b の積となる(式(4))。また、ノード 0~ n は競合ノード数 c_n が等しく、 $P_x = W$ なので式(3)から $\frac{P_n}{P_{max}(n)} \geq 1$ より $S_n = P_{max}n$ なので式(5)となる。さらに、 $P_{max}(y)$ は、ノード y に届いたデータの内、自ノード宛て以外のデータを転送するため、 $P_{max}(x)$ と $T_{max}(0)$ の差となる(式(6))。これら式(4)(5)(6)と式(2)(3)より、式(7)が導ける。

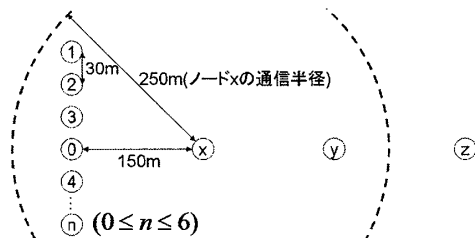


図1: ネットワークポロジ

表2: 設定した通信

送信ノード	プロトコル	通信経路	データ発生量
ノード 0	TCP	$0 \Rightarrow x \Rightarrow y$	T_0
ノード n	UDP	$n \Rightarrow x \Rightarrow y$	W
ノード x	UDP	$x \Rightarrow y \Rightarrow z$	(=2Mbps)

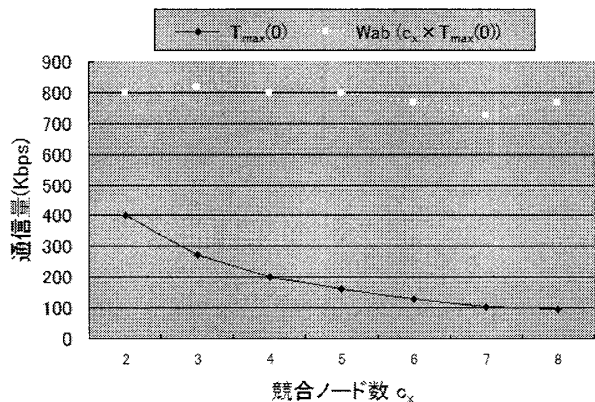


図2: 競合ノード数と最大 TCP データ量

$$T_{max}(0) = P_{max}(0) \times b \tag{4}$$

$$P_{max}(0) = P_{max}(1) = P_{max}(2) = \dots = P_{max}(n) \tag{5}$$

$$P_{max}(y) = P_{max}(x) - T_{max}(0) \tag{6}$$

$$P_{max}(x) = T_{max}(0) = \frac{Wab}{c_x} + \frac{1}{W} \div \frac{Wab}{c_x} \tag{7}$$

4.3 シミュレーション結果

ノード n を 1~6 個間に変化させたときの $T_{max}(x)$ 値変化を図2に示す。式(7)の通り、競合ノード数 c_x が増加すれば $T_{max}(0)$ が減少することがわかる。また、 $Wab(c_x \times T_{max}(0))$ が一定値になることが確認できた。よって、ボトルネックノード x が大量の UDP データを n 個のノードから受信し、1 個のノードにデータを転送する最悪状況下において、遅延なく伝達できる TCP データ量はノード x の競合ノード数 c_x から導けることがわかる。

以上により、本手法を用いることで MANET に大量の UDP データが流れる中で、遅延なく伝達できるデータ量が判明した。これにより、送信ノードのアプリケーションが送信量を調節できれば、TCP を用いた遅延ない重要データの伝達が可能となる。

5. おわりに

本論文では、送信キューを優先度ごとに作りデータを優先度順に送信することで、重要データの遅延欠損をなくす方法を提案した。さらに、シミュレーションを用いて、いかなる状況下でも送信可能なデータ量を求めた。

参考文献

[1] M. Portmann, and A.A. Pirzada, "Wireless mesh networks for public safety and crisis management applications," IEEE Internet Computing, vol.12, no.1, pp.18-25, 2008.
 [2] H. Xiao, K. Chua, J. Malcolm, and Y. Zhang, "Theoretical analysis of tcp throughput in adhoc wireless networks," IEEE GLOBECOM, vol.5, pp.2714-2719, Dec 2005.