

# モバイルノードを導入したトリガ数え上げアルゴリズム

安達 駿 大下 福仁 角川 裕次 増澤 利光

大阪大学大学院情報科学研究科コンピュータサイエンス専攻

**概要** センサネットワークにおいてモニタリングは重要な問題である．代表的なモニタリングの一つに，ネットワーク全体で一定数のトリガが発生したことを検出するトリガ数え上げ問題がある．各ノードではメッセージの送受信に電力を消費する一方で，センサネットワークなどでは消費可能電力が限られていることが多いため，できる限り少ないメッセージ数でこの問題を解くアルゴリズムが必要とされている．そこで本稿では，トリガが正規分布に従って発生するような場合を考え，効率的にトリガ数え上げ問題を解くアイデアを説明し解析を行う．そして最後に今後の課題について述べる．

## 1 はじめに

センサネットワークなどのネットワークにおいて，天候の状態，ログインした端末の数などを監視する分散モニタリングは重要な問題である．その分散モニタリングにおいて，降雨量やログインした端末の数などの発生した事象（トリガ）が一定数（以降，検出トリガ数）を越えたことを検出しユーザに知らせる問題はトリガ数え上げ問題（図 1）と呼ばれ研究が行われている．[1]

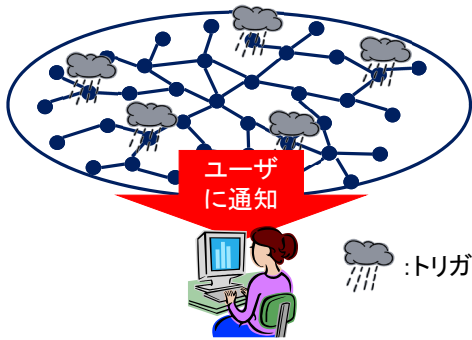


図 1: 検出トリガ数が 5 個の場合の検出例

センサネットワークの場合，各ノードは電力の限られたバッテリーで動作することが多く，そのバッテリーの交換にも多大なコストがかかるため，できるだけメッセージのやりとりを少なくするようなアルゴリズムが望まれている．

そこで文献 [1] では，ノードの数を  $n$ ，検出トリガ数を  $w$  とした時，検出までにシステム全体で交換するメッセージの数（以降，総メッセージ数）の下界が  $\Omega(n \log w)$  であることを示し，総メッセージ数が最適である  $O(n \log w)$  となる集中型アルゴリズムを提案している．しかしこのアルゴリズムは，あるノードがシステム全体を連携させる代表としての役割を果たし，その他のノードはそのノードに従うという集中型制御に基づいているため，各ノードが受信するメッセージの最大数（以降，最大受信数）は総メッセージ数と同じ  $O(n \log w)$  となってしまうことが分かっている．そこで文献 [2] では，乱択アルゴリズムによりメッセージを各ノードに分散して受信させ，総メッセージ数は

最適のまま，最大受信数を高い確率で  $O(\log w)$  に改善することに成功している．

これらのアルゴリズムでは，どのノードでトリガが発生するか，またそのタイミングは予め知ることができないと仮定している．しかし，集中的な降雨などは，過去のトリガ発生情報をもとに推測可能な場合も考えられる．そこで本稿では，トリガの発生確率が事前に分かっている場合（もしくは推測が可能な場合）について考え，より効率的にトリガ数え上げ問題を解くことを考える．具体的には，トリガが正規分布に従い発生するような場合について考え，簡単な検出のアイデアの説明とその解析を行う．そして最後に，今後の研究方針について述べる．

## 2 諸定義

### 2.1 モデルと問題

本稿で考える分散システムは  $m \times m$  のグリッドを想定している．以下が分散システムの定義である．

分散システム  $G = (V, E)$  は  $n = m \times m$  個のノードで構成されている．各ノードは位置する座標  $[0, m-1] \times [0, m-1]$  を表す固有の識別子（以降，ID）を持っており，ノードの集合  $V = \{v_{0,0}, v_{0,1}, \dots, v_{i,j}, \dots, v_{m,m}\}$  と表される．ただし，ノード  $v_{i,j}$  は座標  $(i, j)$  に位置するものとする．また，各ノード  $v_{i,j}$  はリンクを通じて，他のノードと通信を行う．このノード間のリンクの集合を  $E$  とし，本稿ではグリッドを想定しているため  $E = \{(v_{i,j}, v_{i-1,j}), (v_{i,j}, v_{i+1,j}), (v_{i,j}, v_{i,j+1}), (v_{i,j}, v_{i,j-1}) \mid v_{i,j} \in V\}$  となる（図 2）

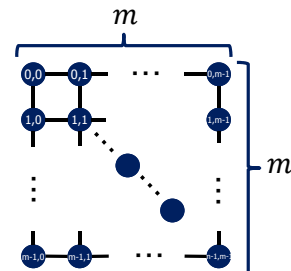


図 2:  $m \times m$  グリッドネットワーク

このような分散システム  $G$  の各ノードで、トリガの発生する座標それぞれが正規分布  $f(x) = \frac{1}{\sqrt{2\pi}\rho} \exp\{-\frac{(x-\mu)^2}{2\rho^2}\}$  に従い発生する状況を考える。(  $x$  はノードの ID,  $\rho^2$  は分散,  $\mu$  は平均を意味している。) 各ノードは正規分布によりトリガが発生することを知っていると仮定しているため、トリガが最も集中して発生するノードの ID( $\mu_x, \mu_y$ ) を知っている。そしてシステム全体で発生したトリガの総数がある与えられた値  $w$  (以降、検出トリガ数) に達したことを、いずれかのノードが検出する問題をトリガ数え上げ問題と呼んでいる。

## 2.2 アルゴリズムの評価尺度

メッセージの送受信には電力を消費するが、センサネットワークなどでは各ノードの消費可能な電力が限られていることが多い。あるノードの送受信するメッセージ数が多数ある場合、そのノードのライフタイムを縮める要因となる。そこでアルゴリズムの性能評価尺度として、次の2つの評価を行うことにする。

- 総メッセージ数 (totalMsg)
  - システム全体で交換されるメッセージの総数
- 最大受信数 (maxRcv)
  - 各ノードが受信するメッセージの最大数

## 3 アルゴリズムと評価

### 3.1 アルゴリズム

本稿の検出アルゴリズムは非常にシンプルである。正規分布が平均値  $\mu$  の付近にデータが集中する分布であることを利用し、そのピークとなるノード  $v_{\mu_x, \mu_y}$  (以降、代表ノードと呼ぶ) が、メッセージによりトリガ情報を集め、システムを代表して発生トリガ数を数えるというものである。各ノード  $v_{i,j} (\neq v_{\mu_x, \mu_y})$  の動作は Algorithm1 の通りである。概要としては、そのノードでトリガが発生した場合、代表ノードへトリガ発生を通知する TRIGGER メッセージを伝達するというものである。この TRIGGER メッセージは、トリガを検出したノードから代表ノードまで最短経路を用いて伝達される。

Algorithm2 は代表ノード  $v_{\mu_x, \mu_y}$  の動作を表している。発生したトリガと各ノードから送られてくる TRIGGER メッセージを数え、その数が検出トリガ数  $w$  を越えた場合にユーザへ通知を行っている。

---

#### Algorithm 1 各ノード $v_{i,j} (\neq v_{\mu_x, \mu_y})$ の動作

---

Constants:

$w$  : 検出トリガ数

$v_{\mu_x, \mu_y}$  : 代表ノード

Function: トリガが発生した場合

< 代表ノード  $v_{\mu_x, \mu_y}$  宛にメッセージ送信 >

1: 自分自身  $v_{i,j}$  宛に TRIGGER メッセージを送信

Function: TRIGGER メッセージを受信した場合

2: if ( $i < \mu_x$ ) then

3:  $v_{i+1,j}$  宛に TRIGGER メッセージを転送

4: else if ( $i > \mu_x$ ) then

5:  $v_{i-1,j}$  宛に TRIGGER メッセージを転送

6: else if ( $j < \mu_y$ ) then

7:  $v_{i,j+1}$  宛に TRIGGER メッセージを転送

8: else if ( $j > \mu_y$ ) then

9:  $v_{i,j-1}$  宛に TRIGGER メッセージを転送

---



---

#### Algorithm 2 代表ノード $v_{\mu_x, \mu_y}$ の動作

---

Constants:

$w$  : 検出トリガ数

Variables:

$counter \leftarrow 0$  : カウンタを初期化

Function: TRIGGER メッセージを受信した場合

1:  $counter++$

2: if ( $counter == w$ ) then

3: 検出トリガ数  $w$  に達したことをユーザに通知

Function: トリガが発生した場合

4:  $counter++$

5: if ( $counter == w$ ) then

6: 検出トリガ数  $w$  に達したことをユーザに通知

---

### 3.2 シミュレーションによる評価と考察

グリッドのサイズ  $m = 10, 20, \dots, 200$  について総メッセージ数、最大受信数について、シミュレーションにより評価を行った。シミュレーションの際のパラメータは表1の通りである。

検出トリガ数	100000
平均 $\mu$	$\lfloor m/2 \rfloor$
分散 $\rho$	1.0
代表ノードの ID( $\mu_x, \mu_y$ )	$(\lfloor m/2 \rfloor, \lfloor m/2 \rfloor)$

表 1: シミュレーションのパラメータ

総メッセージ数、最大受信数は図 3,4,5 のようになった。

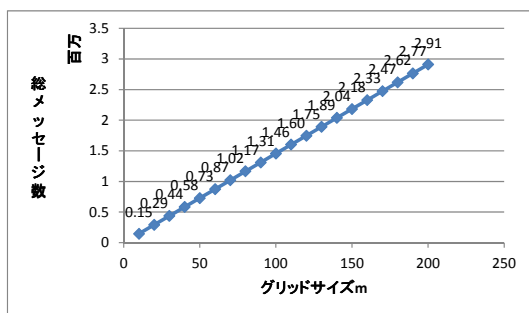


図 3: シミュレーション結果：総メッセージ数

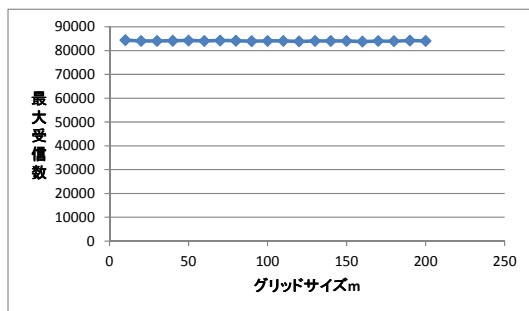


図 4: シミュレーション結果：最大受信数

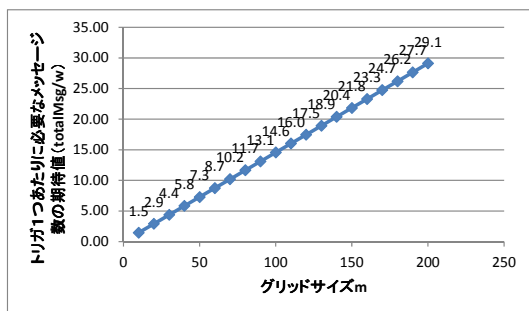


図 5: トリガを代表ノードに知らせるのに必要なメッセージ数の期待値

図 5 より，トリガ 1 つに対するメッセージ数の期待値はグリッドのサイズ  $m = \sqrt{n}$  に比例していることが分かる．トリガ数上げ問題の下界 ( $\Omega(n \log w)$ ) が  $n$  に比例したものであったことを考えると，改善が期待できる結果となっている．その一方で，図 4 のように，1 つのノード（代表ノード）にメッセージが集中してしまう欠点があることも分かる．

## 4 今後の課題

本稿では，正規分布に従いトリガが発生する場合についてシンプルなアルゴリズムのシミュレーションを行い，メッセージ数の評価を行った．

今後の課題としては，図 4 から分かるように，本稿の手法は 1 つのノード（代表ノード）にメッセージが集中してしまう欠点の改善が挙げられる．そこで近年注目されているドローンのようなモバイルノードの導入を検討したいと考えている．モバイルノードは，電

力の制限を考慮しなくてよいという長所がある一方で，通信範囲外のノードとの通信には大幅な遅延が発生するという欠点がある．そこでこのモバイルノードが代表ノードの役割を果たすことで，効率的なアルゴリズムを考案できるのではないかと考えている．さらに，集中的にトリガが発生する場所が変化していくような（トリガ発生 of the normal distribution of the peak changes in such a way）場合についても考え，モバイルノードの制御を含めたアルゴリズムを検討していきたいと考えている．

## 参考文献

- [1] Rahul Garg, Vijay K.Garg, and Yogish Sabharwal. Scalable algorithms for global snapshots in distributed systems. In *Proceedings of the 20th annual international conference on Supercomputing(ICS)*, pp. 269–277, 2006.
- [2] Venkatesan T. Chakaravarthy and Yogish Sabharwal. An optimal decentralized algorithm for the distributed trigger counting problem. Technical report, IBM Research - India, 2011.