Vol.2020-CDS-29 No.4
2020/12/3

送信量解析を用いたセンサネットワーク異常予測手法の提案

加藤 亜慧[1]　中濱 広夢[1]　清原 良三[2]　斎藤 正史[3]　寺島 美昭[1]

概要：本稿では, アドホック通信を用いるセンサネットワークを対象に, センサを搭載した無線端末のデータ送信量の減衰と伝搬傾向を解析し, 通信が完全途絶する前に通信異常を検出する異常予測手法を提案する. アドホック通信は, 利用環境に応じて接続性を確保するきめ細かいチューニングや, 帯域幅の狭さ、通信の安定度の低さがあるため, 有線環境にて用いられるパケットキャプチャ情報を用いた共通手順による解析が難しい. 提案手法は, センサ端末の送信量変化のみを用いることで, 検出精度は低いものの汎用性を備えた異常予測を実現する. 本報告では, ネットワークシミュレータを用いた提案手法の理論検証も報告する.

A Study of Anomaly Prediction Method in Sensor Network Using Terminal Transmission Analysis

1. はじめに

近年, 被災地における迅速な環境観測に無線通信端末を搭載した複数のセンサを展開するセンサネットワークが注目され [1], 例として, 車や無人航空機「ドローン」の活用が挙げられる. そのようなセンサを用いた環境観測では, 一般的な基地局を介した無線通信だけでなく, センサ端末間で相互通信を行い情報共有する無線アドホック通信の利用が考えられている [2][3]. しかし、無線アドホック通信では無線特有の劣悪な電波伝搬や雑音の多さに加え、通信機器の故障など通信異常発生のリスクが高い。従来の通信異常検出では異常のある端末が完全途絶するまで通信を続け, タイムアウト後に端末を特定してチャネルの変更やセンサの再配置といった対応を行う. このような対処は通信が一瞬で途絶する場合には有効である。一方, 徐々に減衰し途絶する場合においては、タイムアウトまで待たず, 減衰している段階で異常を予測することが望ましい。しかしながら, アドホック通信は前述の通り通信安定度の低くく, 利用環境に応じて接続性を確保するきめ細かいチューニングや帯域幅の狭さがあるため、有線環境にて用いられるパケットキャプチャ情報を用いた共通手順による解析が難しい [4]. このことから，本研究では，センサ端末の送信量変化のみを用いることで、検出精度は低いものの汎用性を備えた異常予測手法の検討を行っている. 監視システム構成を図 1 に示す.

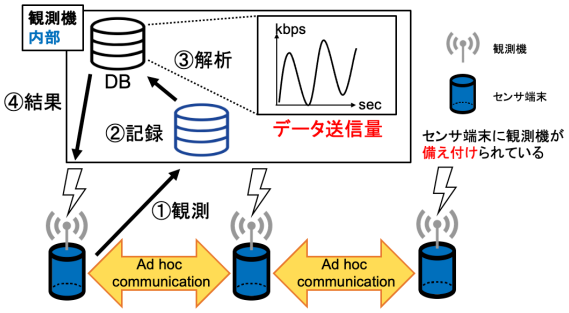

図 1　監視システム構成

データ送信量を記録する観測機は各センサ端末ごとに備え付けられ, 自端末のみ記録・解析を行う. 高精度にデータ送信量を記録し, 各センサ端末それぞれで自律的に通信異常を検出する目的で本構成を採用した. 通信は双方向で行っている. この構成から得られたデータ送信量を用いて提案する異常予測アルゴリズムでは, 通信異常端末の予測検知を行う. 検知方法は, 自律的な異常予測を行う観点から他端末との送信量の比較を行わないため, 自己比較として事前収集した正常時のデータ送信量に基づく値 (基準値) を作成し, これに対する観測データ送信量の変化量 (差分) か

1　創価大学大学院 工学研究科
Graduate School of Engineering, Soka University
2　神奈川工科大学 情報学部
Faculty of Information Technology, Kanagawa Institute of Technology
3　金沢工業大学 情報フロンティア学部
College of Informatics and Human Communication, Kanazawa Institute of Technology

ら通信異常の判断をする. この基準値は運用する通信環境に合わせ動的に決定できるという利点もある. 本研究では, ネットワークシミュレータを使用した提案手法の理論検証を行う.

以降,2 章では本稿の課題を述べる. 3 章では提案アルゴリズムの詳細,4 章ではネットワークシミュレータを用いた実験内容と提案アルゴリズムの有用性の検証結果を述べた後,5 章で全体のまとめとする.

2. 目的と課題

本研究の目的は, 各センサ端末それぞれが自身から得られた単体のデータ送信量変化のみを用いて自律的に通信異常を正確でなるべく事前に検知することである. 通信異常には複雑で様々なモデルが考えられるが, 本アルゴリズムの検出対象とする通信異常モデルは図 2 に示すように徐々に (直線的に) 通信が減衰し、途絶する場合としている. 詳細については,4 章の実験章に記述する. 図 2 のモデルの場合, 通信途絶すれば必ず異常を検知できるため精度を高めつつ, なるべく事前に検知することが重要な課題となる.

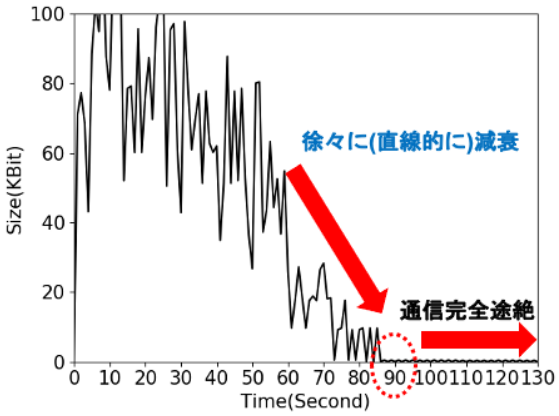

図 2 通信異常モデル例

精度に関して問題となるのが通信異常端末の影響による正常端末の誤検知である. アドホック通信の特性上単方向通信の場合, 図 3 のように伝搬するためネクストホップの隣接端末に異常端末と同じ傾向が見られ, 一意に判別できない (異常端末のみを検知できない) 問題が存在した. そこで双方向通信を行い, 通信を掛け合わせることで自端末のデータ送信量のみで観測機が自身のセンサ端末の通信異常を正確に判断しようと考えた. これにより正常端末への影響をなくすとまではいかなくとも, 異常端末にのみ現れる特徴を顕在化できる. 提案する異常予測アルゴリズムは, この双方向通信が対象とするセンサネットワークに制御通信プロトコルとして組み込まれている前提で開発を行った.

検知を行う際, 取得したデータ送信量をそのまま扱うと通信環境によっては突発的な値の上昇や下降のような雑音が頻発し、異常予測の精度が落ちてしまう. これは高精度を求める場合, 検知を遅延させることを意味する. そのため, アルゴリズムにおける工夫として, 取得したデータ送信量を通信の不安定さに合わせ平滑化し, 雑音を除去して異常予測を行う.

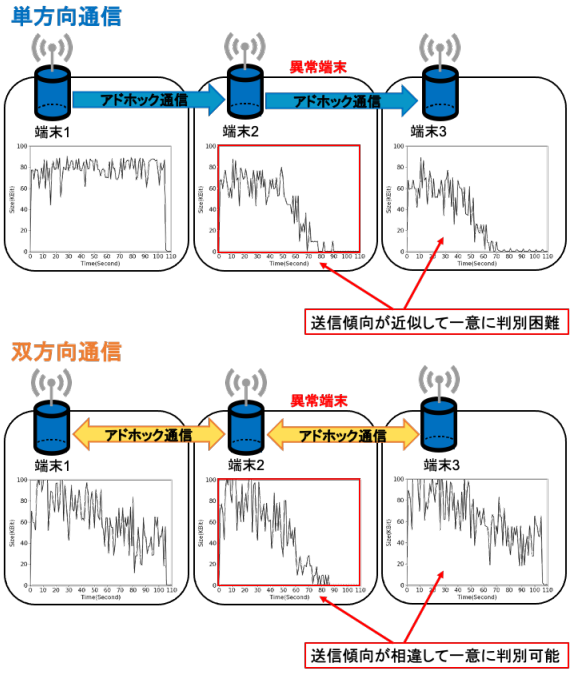

図 3 単方向/双方向通信における波形の傾向

3. 異常予測アルゴリズムの提案

3.1 異常予測アルゴリズムの概要

本稿では, センサネットワークを対象に, 観測機が自身のセンサ端末から得られるデータ送信量のみで自端末における自律的な通信異常の予測検知を実現するための異常予測アルゴリズムを提案する. 図 4 に異常端末予測検知の手順を示す.

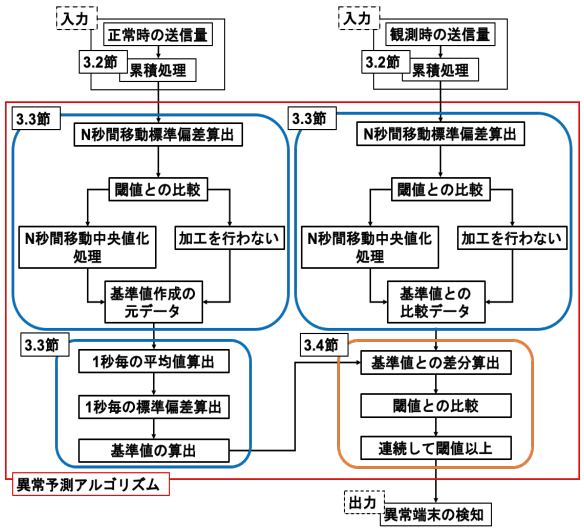

図 4 異常予測アルゴリズムのフロー

提案アルゴリズムは, 入力に本研究で用いる監視システム構成によって得られた観測時と正常時の各センサ端末それぞれのデータ送信量のみを用いる．取得したデータ送信量は無線通信であることから非常に不安定であり、突発的な値の上昇や下降のような雑音の影響を受け, 通信異常による減衰なのか通信の不安定性が原因なのか判断が難しい場合がある. そのため, データ送信量を通信の不安定さにあわせて移動中央値で平滑化し, 雑音の除去を行う. また、運用する通信環境に合わせ自律的な異常予測を実現するため, 自己比較として事前収集した正常時のデータ送信量に基づく値 (基準値) を作成し, 観測データ送信量の変化量 (差分) から通信異常の判断をする.

3.2 観測データの前処理

提案アルゴリズムによる解析の前処理として, 取得したデータ送信量を 1 秒ごとの値に累積処理する. これにより, 単位時間当たりの通信品質の劣化が顕著となる [4]. 図 5 にアルゴリズムを示す.POINT1 で単位時間当たりのデータ送信量の累積処理を行っている.

入力:取得送信量(*GtN*)
出力: 1 秒ごとの累積送信量(*CtN*)

N=Node 番号
t=時間(秒)
MAXTime=シミュレーション時間

for(N=1; N<=端末数; N++){
  for(t=0; t<= MAXTime; t++){
    $CtN=\sum_{n=t-1}^{t} GtN$
  }
}

POINT1:累積送信量の算出

図 5 累積処理のアルゴリズム

3.3 基準値の作成

本稿では, 自律的な異常予測を実現するため通信異常の自己判断基準として正常時のデータ送信量から作成する基準値について説明する. 提案アルゴリズムでは、基準とする送信量とどの程度観測したデータ送信量に差があるかによって異常を判断する. 詳細については 3.3 節で述べる. 通信の不安定性により正常時でも一時的な送信量の落ち込みが考えられることや運用環境によって通信状況が多数あることから予め決めた一様な基準ではなく, 運用時に得られた正常な送信量から基準を作成することとしている. また、同様の理由で一定期間の移動標準偏差を取り, 通信の不安定性を数値化することで決められた閾値以上の値の場合は、移動中央値を用いて平滑化処理を行う.

作成方法は正常時のデータ送信量のケース数が N 個あった場合, まず先程述べた移動標準偏差を算出して平滑化処理を行う. この処理は標準偏差の閾値 (標準偏差判別閾値) によって平滑化の割合が変化し, 通信の状況次第で全く行わないことも全て平滑化することも可能である.1 秒ごとに N 個の値の平均値-2 標準偏差をとる. これは平均値 ± 2 標準偏差の間に約 95 %のデータが含まれる傾向を利用している. そのため 1 秒ごとに基準値は変動する. 標準偏差判別閾値別の基準値作成のイメージを図 6 に示す.

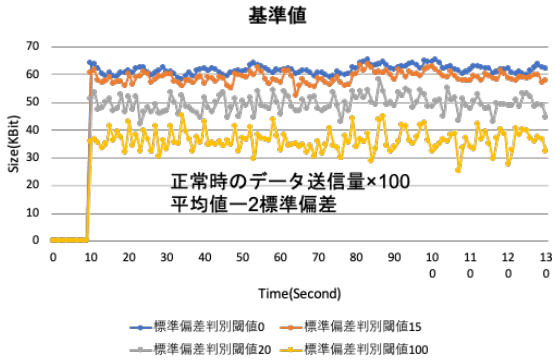

図 6 基準値作成のイメージ

図 7 に本アルゴリズムを示す．Point1 は Pi 秒間の移動標準偏差を計算している. 移動標準偏差を計算することでその区間のばらつき, つまり通信の不安定性を数値する.Point2 では計算した標準偏差が閾値より大きいかよってから平滑化処理を行うかそのままの値を扱うかを判断する. 平滑化する場合は Pi 秒間の移動中央値を計算し, 基準値作成の元データとする.Point3 では元データのケース数に応じて 1 秒ごとの平均-2 標準偏差を計算し, 基準値を作成する.

入力:正常時の 1 秒毎の累積送信量(aCtN)
出力:基準となる累積送信量(RCtN)

N=Node 番号
t=時間(秒)
MAXTime=シミュレーション時間
S=シミュレーション回数
aCtN =正常時の 1 秒ごとの累積送信量
aC(t-Pi)N'~aCtN' =aC(t-Pi)N~aCtN を小さい順に並べた値

for(S=1; S<=最大シミュレーション回数; S++){
  for(N=1; N<=最大端末数; N++){
    for(t=1; t<= MAXTime; t++){
      $MaCtN=\frac{1}{Pi}\sum_{n=t-Pi+1}^{t} aCtN$　(MaCtN は 1 秒毎の移動平均累積送信量,Pi は閾値)
      $MaStN=\sqrt{\frac{1}{Pi}\sum_{n=t-Pi+1}^{t}(aCtN-MaCtN)^2}$　(MaStN は MaCtN の移動標準偏差)

POINT1:ばらつき度合いの算出

      if(MaStN>X)
        ACtN=1/2(aC(t-Pi/2)N'+aC(t+1-Pi/2)N')
      else
        ACtN=aCtN　(ACtN は基準値作成に用いる 1 秒毎の累積送信量)

POINT2:累積送信量の一部平滑化

    }
  }
}

for(N=1; N<=最大端末数; N++){
  for(t=1; t<= MAXTime; t++){
    $bCtN=\frac{1}{S}\sum_{n=1}^{S} ACtN$　(bCtN は正常時の 1 秒毎の平均累積送信量)
    $bStN=\sqrt{\frac{1}{S}\sum_{n=1}^{S}(bCtN-ACtN)^2}$　(bStN は bCtN の標準偏差)
    RCtN=bCtN−2bCtN　(RCtN は基準値)
  }
}

POINT3:基準値の作成

図 7 基準値作成のアルゴリズム

3.4 差分検知

本稿では, 観測時のデータ送信量の平滑化処理と基準値との差分を取ることによる異常端末の検知方法について説明する. 平滑化処理の手順は基準値作成時と同様である. そして処理を行った観測データ送信量と基準値の差分をとり, 閾値より値が大きくとなった場合通信異常端末として検知する. この際平滑化処理を行っている場合でも一瞬差分が閾値を超えてしまう危険性を考慮して連続検知回数という閾値を加えて導入している. これにより設定した連続検知回数以上連続で差分が閾値を上回った場合に限り通信異常端末として検知する. 図 8 に差分検知のイメージを示す.

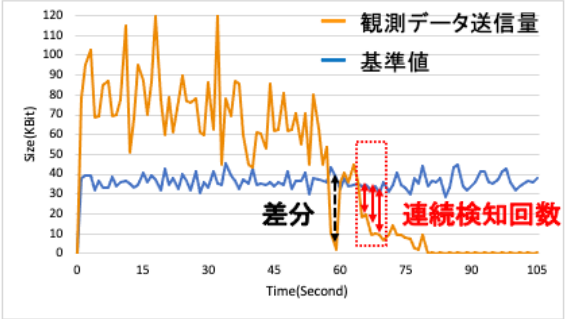

図 8 差分検知のイメージ

図 9 に本アルゴリズムを示す. Point1,2 は 3.2 節で説明しているため省略する.Point3 では処理済みの観測データ送信量と基準値の差分を取り, 閾値を上回った場合連続検知回数が 1 足される. そして,Point4 で連続検知回数が設定した値以上となった場合通信異常端末として検知をする. なお, 途中で差分が閾値を下回り, 連続検知が中断された場合連続検知回数は初期化される.

```
入力:観測時の 1 秒ごとの累積送信量(CtN)
出力:故障端末 A

N=Node 番号
t=時間(秒)
MAXTime=シミュレーション時間
S=シミュレーション回数
C(t-Pi)N'-CtN' =C(t-Pi)N-CtN を小さい順に並べた値
B=検知回数

POINT1:ばらつき度合いの算出

POINT2:累積送信量の一部平滑化

for(N=1; N<=最大端末数; N++){
 for(t=1; t<= MAXTime; t++){
  if(RCtN−TCtN>X){   (X は閾値)
   検知回数 B に 1 を追加                    POINT3:観測値との差分検知
   if(B=Y){   (Y は閾値)
    NodeN は相対減衰端末
    A に追加
   }
  }
  else{
   検知回数 B を初期化
  }                                          POINT4:連続Y秒間検知した端末抽出
 }
}
```

図 9 差分検知のアルゴリズム

4. 検証実験・評価

4.1 実験手順

提案アルゴリズムにおける異常予測検知の精度と早さの確認および閾値による変化の理論検証を行う目的で実験を行った. 実験では, 図 10 に示す双方向通信を前提としたネットワーク構成を用いている. 実際のセンサネットワークの運用環境を想定し, 隣接端末との影響を検証するため 3 × 3 のメッシュ型ネットワーク構成を設計した. 通信に関与しない端末は常に 0〜数 KBit のデータ送信を行う. パラメータは表 1 のとおりである.

トポロジー 1 は異常端末が送信元に挟まれている状況である. トポロジー 2 は方向転換が加わり, 異常端末は他端末に囲まれている. トポロジー 3 は異常端末が送信元の隣接せず, 距離が長くなる. それぞれの端末の初期配置は変化しない. 端末間距離である 180m は 1 ホップで通信が行える限界値として予備実験で導き出した.

表 1 メッシュ型ネットワーク構成のパラメータ

項目	設定値
端末数	9 端末
端末間距離	180m
トポロジー 1/2/3 の端末数	3 端末/4 端末/5 端末
トポロジー 1/2/3 の送信元端末	端末 1,3/端末 1,6/端末 1,9
トポロジー 1/2/3 の異常端末	端末 2/端末 5/端末 7

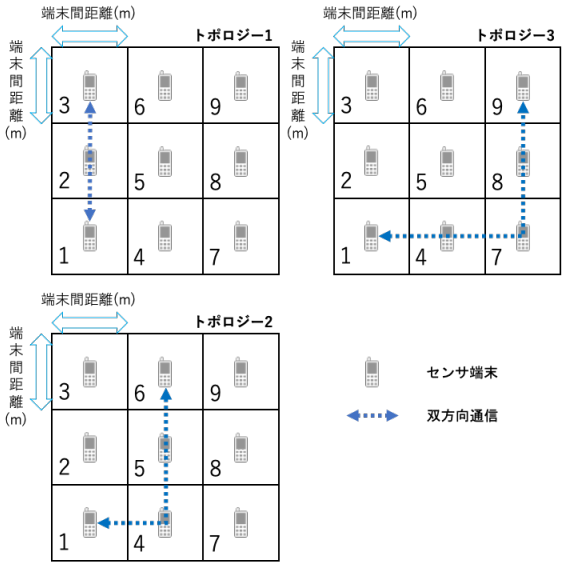

図 10 メッシュ型ネットワーク構成

入力とするデータ送信量の作成にはネットワークシミュレータ QualNet[5] を使用している. 提案手法ではセンサ端末に観測機を備え付け解析を行うため, シミュレーションでは観測機を設置せず, 実験では各端末から送信されたパケットを直接解析し検知を行っている. シミュレーション設定は表 2 に示す設定の通りである.

表 2　シミュレーション設定

Simulator	QualNet7.4
Simulation Time	130s
Number of trials	100seed
Radio Type	802.11b
Data Rate	2Mbps
Frewuency Band	2.4GHz
Application	Constant Bit Rate

ルーティングプロトコルの仕様は，MANET(Mobile Ad hoc NETwork) における代表的なプロアクティブ型のプロトコルである Optimized Link State Routing INRIA(以下 OLSR) を用いた. 決められた端末内での制御通信を前提としていること, 異常端末を確実に検知する観点から静的ルーティングで通信を行っている. 各種パラメータのチューニングは行っていない. 送信元は 0.1 秒ごとに 1024byte のパケットを送信している．今回の実験では，閾値を複数パターン用意し, 検証を行った. 表 4 に設定したアルゴリズムの各閾値を示す. 標準偏差の判定閾値において,0KBit は送信量のばらつきが 0KBit より大きい場合, つまり全てのデータ送信量を平滑化することを意味し,15KBit・20KBit は 1 部の平滑化を意味する. 一方で 100KBit は全く平滑化を行わない. また, 差分の大きさが差分閾値を 1 度上回った場合と複数回上回った場合で比較するため連続検知回数を 1 回と 3 回としている.

表 3　異常アルゴリズムの閾値

項目	設定値
移動標準偏差の間隔	10 秒
標準偏差の判定閾値	0/15/20/100KBit
移動中央値の間隔	10 秒
観測データと基準値の差分閾値	5KBit
連続検知回数	1 回/3 回

4.2　異常モデル

アルゴリズムの異常検知の精度と速さを検証するため, 通信が徐々に (直線的に) 減衰し、途絶する異常モデルをその減衰の傾斜角度ごとに 3 モデル定義した. 実験では, 単に傾斜モデルを作成するのではなく, シミュレーション上で端末を真上に打ち上げ他端末との距離を徐々に空けていくことで実際に通信の減衰・途絶を模擬している. 図 11 に各異常モデルを示す.

異常モデルはシミュレーション開始から 15 秒後に発生する設定である. 減衰の速度を表す平均通信途絶時間は, 同じ異常モデルであってもシミュレーションのケースにより完全途絶にばらつきがあることから,2 秒間送信量が 1KBit 未満となった場合通信途絶と定義して 100 ケースの平均を計算した.

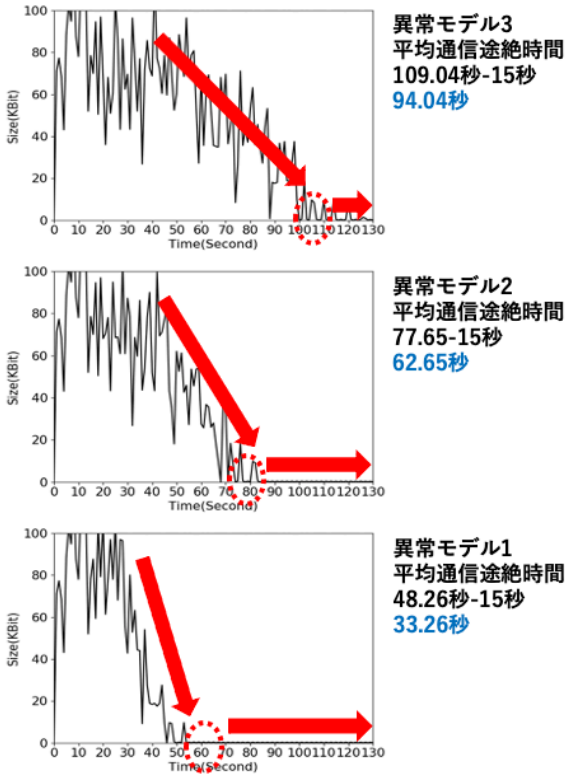

図 11　異常モデル

4.3　評価方法

評価方法について述べる. まず検知の時間と検知数の評価について説明する. 図 12 に示すグラフが検知時間ごとの検知数の分布を表す実験結果の 1 例である. 実験結果を見ると異常発生開始時間である 15 秒やその直後の時間において, すでに数回検知が行われている事がわかる. しかし, これは偶然もしくは誤検知である可能性が高い. 一方で時間的にどこからを有効とするか決めることは難しい. そのため, ある程度検知時間の分布がまとまると考え,10 より多く検知した場合それ以降を現実的な値として有効範囲と定義し, 評価することとした.

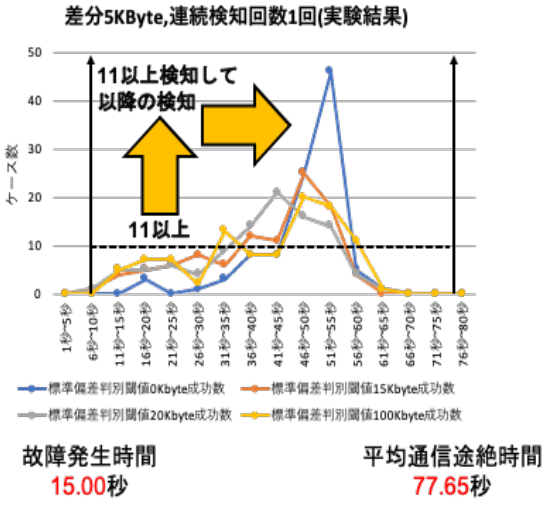

図 12　実験結果

評価結果が図 13 のグラフである. 時間ごとに検知結果の累積を行い, 検知の早さにおける検知数を表している. 次節ではこのグラフを元にそれぞれの評価結果を述べる.

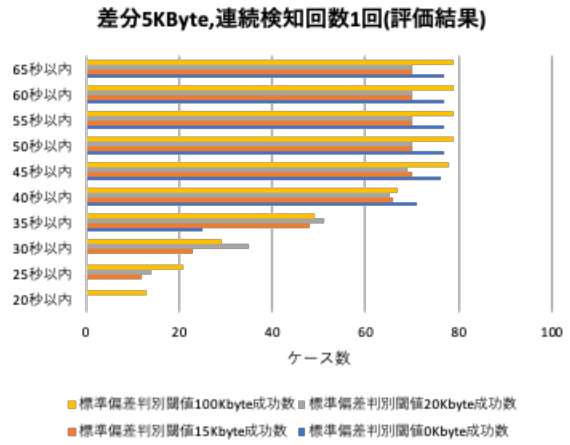

図 13 評価結果

次に検知の失敗について説明する. 本研究で扱う異常モデルは,2 章でも述べたように必ず検知が可能である. そのため, 一般的な異常端末を正常端末として判断する誤検知は存在しない. 加えて,1 台の異常端末を予測検知する観点から本来の異常端末より先に正常端末を異常端末と判断した場合に検知の失敗と定義した.

4.4 評価結果

異常予測アルゴリズムの評価結果を異常モデルごとに示す. 本稿では, 図 11 に示したトポロジーの内トポロジー 3 の評価結果を取り上げる.

提案アルゴリズムに従い, 標準偏差判定閾値 (以降判別閾値) ごとに加工 (平滑化) を行った正常時データ送信量から基準値を作成し, 同一の判別閾値で加工した観測データ送信量との差分から, 連続検知回数を上回った端末を通信に異常のある端末として検知する. 異常モデル 1 において図 14 は判別閾値別に何秒以内に何ケース異常端末を検知できるか (検知成功数), 図 15 は検知失敗数を表したグラフである.

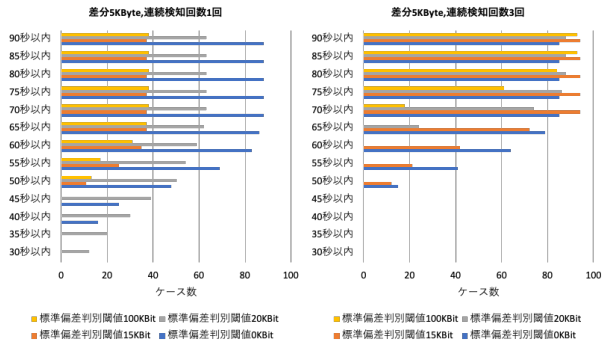

図 14 異常モデル 1 の検知成功数評価結果

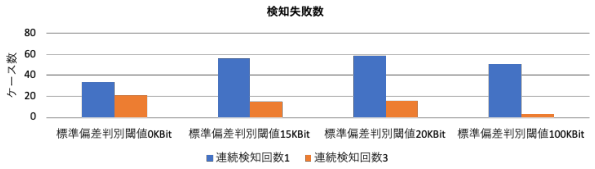

図 15 異常モデル 1 の検知失敗数評価結果

検知成功数では連続検知回数 1 回において,30 秒以内に 12 ケース検知した判別閾値 20KBit の場合が最速である. 一方,70 秒以内に 88 ケース検知した判別閾値 0KBit の場合が最速で最多の検知となった. 連続検知回数 3 回においては,50 秒以内に 15 ケース検知した判別閾値 0KBit の場合が最速である.50 秒以内に 12 ケース検知可能な判別閾値 15KBit は最終的に 70 秒以内に 94 ケースで最速最多となった. その結果, 異常モデル 1 では, 平均して異常発生から 94.04 秒後に通信途絶しているため, 最大で約 8 割から 9 割予測検知可能であることを確認できた.

検知失敗数については連続検知回数 1 回では,34 ケースと判別閾値 0KBit が最も少なくその他は同程度である. 連続検知回数 3 回にした場合は, 判別閾値 100KBit が 3 ケースで最小となり, 判別閾値 0KBit が 21 ケースで最大となる.

次に異常モデル 2 における評価結果について図 16 に検知成功数を, 図 17 に検知失敗数を示す.

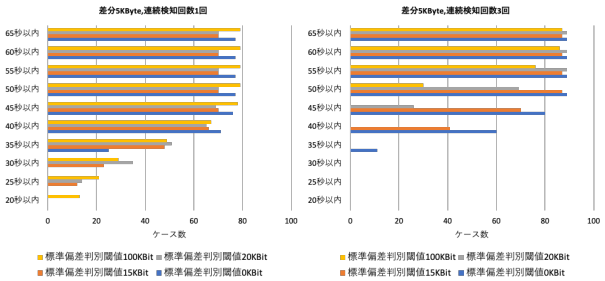

図 16 異常モデル 2 の検知成功数評価結果

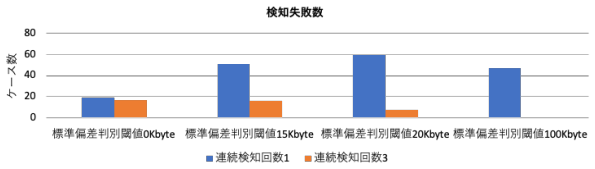

図 17 異常モデル 2 の検知失敗数評価結果

検知成功数では連続検知回数 1 回において,20 秒以内に 13 ケース検知した判別閾値 100KBit の場合が最速である. その後 40 秒以内においては 71 ケース検知した判別閾値 0KBit が最大となり, その後判別閾値 100KBit が 79 ケース検知し, 最大となる. 連続検知回数 3 回では, 反対に判別閾値 0KBit が 35 秒以内に 11 ケース検知して最速となり,50 秒以内まで最大である. 以降判別閾値 20KBit も同数となる. 結果, 異常モデル 1 より減衰が急な異常モデル 2 におい

ても, 平均して異常発生から 62.65 秒後に通信途絶していることから, 同じく最大で約 9 割予測検知可能であることを確認できた.

検知失敗数では判別閾値 0KBit の連続検知回数 1 回/3 回における値は同程度で, 前者では最小, 後者では最大となる. 後者の場合平滑化の度合いを下げていくごとに値が低下していくことが分かる.

次に異常モデル 3 における評価結果について図 18 に検知成功数を, 図 19 に検知失敗数を示す.

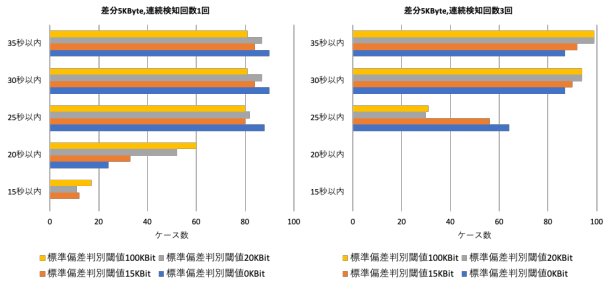

図 18 異常モデル 3 の検知成功数評価結果

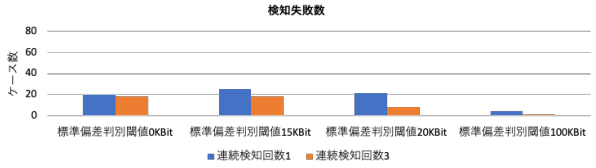

図 19 異常モデル 3 の検知成功数評価結果

検知成功数では連続検知回数 1 回において, 最速である 15 秒以内に判別閾値 100KBit が 17 ケース検知し, その区間で最大となった. その後 25 秒以内の区間で判別閾値 0KBit が追い抜き, 最終的に 90 ケース検知して最大となる. 連続検知回数 3 回では, 初めこそ 25 秒以内に判別閾値 0KBit が 64 ケース検知し最大となったが,30 秒経過以降逆転し, 判別閾値 20KBit/100KBit が同数で最大となる. 結果, 異常モデル 2 よりさらに急激に減衰する異常モデル 3 においても判別閾値により大きく変動するものの同じく最大で約 9 割程度予測検知可能であることを確認できた.

検知失敗数では判別閾値 100KBit における連続検知回数 1 回/3 回の値はともに最小となり, 後者の場合判別閾値 0KBit/15KBit は同じ値なもののその後平滑化の度合いを下げていくごとに値が低下する結果となった.

4.5 考察

以上の検証実験により, 複数のセンサ端末の観測環境を模したネットワークにおいて減衰の傾きの違う 3 つの異常モデルいずれでも, 端末の通信異常を完全途絶前に予測検知可能であると検証できた. 傾向として連続検知回数を 1 回にした場合早期に検知が可能だが, 検知失敗数が上昇する. 一方, 連続検知回数を 3 回にした場合 10 秒から 20 秒ほど検知が遅延するが失敗数は低下する. また, 異常モデル 1,2 の場合連続検知回数が 1 回の場合 1 部もしくは全て平滑化することで検知の早期化と高精度化を図れる. しかしながら, より急激に減衰する異常モデル 3 でこの傾向が逆転し, 全く平滑化を行わない場合の方が検知の早期化と高精度化を図れることが見て取れる.

5. まとめ

本稿では，アドホック通信を用いるセンサネットワークを対象に，データ送信量解析を用いて通信が完全途絶する前にセンサ端末の通信異常を検出する異常予測手法を提案した．提案手法における異常予測アルゴリズムでは, 正常時のデータ送信量から基準値を作成し, 基準値に対する相対的な変化量から通信異常の判断を行う. また, ネットワークシミュレータ QualNet を用いて入力とするデータ送信量を作成して実験を行い, 提案の有用性を検証した. 本検証によりアルゴリズムでは異常端末の通信完全途絶前の予測検知が可能なことが分かった. 実験結果はアドホックネットワークのスケールを含めたシミュレーション設定と異常モデル, 閾値に対する依存度高いため, 今後さらなる検証と検知の早期化, 高精度化に取り組む.

参考文献

[1] 石川久嗣, 横田裕介, 大久保英嗣, “自律飛行体による無線センサネットワークのための移動スケジューリング手法,” 信学技報 (知的環境とセンサネットワーク), Vol.114, pp.55-60 (2015).
[2] Md. Hasan Tareque, Md. Shohrab Hossain, Mohammed Atiquzzaman “On the Routing in Flying Ad hoc Networks” (2015).
[3] 松井進, “アドホックネットワークの実用化に向けた課題と実用化動向”, 日本信頼性学会誌 第 34 巻,pp.532-539, (2012).
[4] 福岡宏一 他, “データ送信量解析を用いたアドホックネットワーク動作推定方式の評価”, 情報処理学会論文誌, 5W-03 ,3-157.3-158 (2019).
[5] QualNet Network Simulator Software (https://web.scalable-networks.com/qualnet-network-simulator-software)