エッジコンピューティングにおける通信切断を考慮した通信ライブラリの試作

2018SE056 中野克紀
指導教員：宮澤元

1　はじめに

近年，IoT(Internet Of Things) の発展により，日常生活の様々なものがインターネットに接続されるようになった．これによりリソースが十分ではない IoT 機器でもデータを処理能力のある機器に送信し，処理をしてもらうことで今までは実現できなかった IoT 機器に対して高度な操作を行うことが可能になった．インターネットとは無縁に見える農業においても，温度や湿度といったデータを測定する各センサや，農薬を散布するドローンなどの機器がインターネットに接続されるようになっている．農場に設置されたセンサは温度や湿度といった農場のデータを計測し，インターネット上のサーバに送信する．利用者はスマートフォンやパソコンからこのサーバに接続し，農場にいなくても温度や湿度の状況をリアルタイムに確認することができる．

しかし，日本は地震や台風とはじめとする自然災害の多い国であり，自然災害によってネットワーク設備がダメージを受けるとセンサとサーバとの通信が正常にできなくなる場合がある．利用者はセンサから連続して得たデータを閲覧し，水や肥料をどの程度与えるのかを判断するので農場からのデータが通信ができない理由で途切れ途切れになるのは望ましくない．また，一般に利用されるセンサはリソースが十分ではないことが多く，データを保持し続け，後に再送信することは難しい．

本研究の目的は，クライアント近くのエッジネットワークがクラウドから一時的に切断されてもアプリケーションが動作し続けられるような通信基盤を実現することである．エッジネットワーク上のホストが通信データを一時的に保持することで，計算リソースを十分に持たないデバイスをクライアントとして用いる場合でも，クラウドとエッジネットワークの接続状況にかかわらずアプリケーションを動作させ続けることができる．

本研究では，クラウドとエッジネットワーク間の通信が一時的に切断されていても，通信状態を保持し続けられる通信ライブラリを提案する．通信切断時はエッジネットワーク上に設置したブローカに通信データを保存し，再接続時にこの通信データを取り出して再送信を行う．これにより，クラウドとエッジネットワーク間の通信が一時的に切断されても，アプリケーションが特別な処理を行うことなく，クライアントからのデータをサーバに損失なく送信できる．

本研究の研究課題は，通信切断時にブローカを利用することで通信切断時でも通信できているように動作する通信ライブラリの作成と簡単な Web アプリケーションを用いた通信ライブラリの有効性の確認の 2 点である．

2　関連研究

福田らは不意の通信切断を考慮した WebSocket フレームワークを提案している [1]．サーバ側にデータベースを設置し，サーバからのクライアントへの通信 (Push 通信) データをデータベースに一時保存することで不意の通信切断を抑制している．

本研究では，IoT 機器のようなリソースの十分でないデバイスがサーバにデータを送信する状況を前提としており，福田らが想定している SNS のような Web アプリケーションを対象とした手法を直接適用することはできない．

3　提案手法

クライアントである IoT 機器からクラウド上のサーバへの送信データを，通信切断時にはブローカを介して送信させることで通信できているように動作する通信ライブラリを提案する．図 1 に示すように，クライアントを含むエッジネットワークはクラウドのサーバと通信できないが，各 IoT 機器はエッジネットワーク上のエッジサーバとは通信できる状況を前提とする．クライアントはサーバに直接送信できなかったデータをエッジサーバ上のブローカに送信する．ブローカはこのデータをサーバとの再接続後にサーバへ再送信する．これによってクライアントは通信切断時でもデータの損失無く送信を行うことができる．クライアントがデータを送信できなかった時，クライアントが送信できなかったデータを保持し続け，再接続後に再送信することでも通信切断時のデータの損失を無くすことができるが，十分なリソースを持たない IoT 機器にデータ再送信を行わせることは望ましくない．

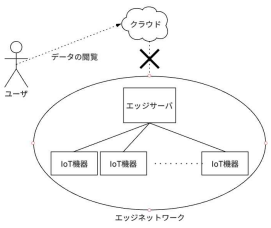

図 1　通信切断の状況

提案ライブラリを用いたクライアントのデータ送信動作を図 2 に示す．クライアントがデータを送信できない場合，クライアントはエッジネットワーク上のブローカに

データを送信する．ブローカはサーバとの接続が回復した時にクライアントから送信されたデータをサーバに再送信する．

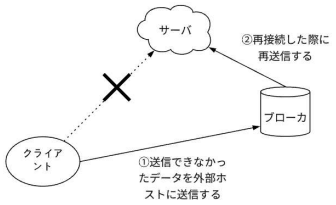

図 2 提案ライブラリよるクライアントのデータ送信

4 実装

HTTP 通信部分に libcurl を，エッジネットワーク上のブローカとして Apache Kafka を用いて提案ライブラリを実装した．本ライブラリはインタフェースとして図 3 に示す myhttp 関数を提供する．myhttp 関数の引数に通信先 URL を指定すると指定した URL に対して http 通信を行い，サーバからの返信を待つ．サーバとの通信ができない場合，通信データをブローカに送信する．本来であればブローカが再送信の処理を行うが，今回は実装を簡単化するために，ブローカは通信データの保持だけを行う．クライアントはサーバとの再接続を試み，成功した場合，ブローカから通信データを取り出し，サーバに再送信する．

```
int myhttp(char *url);
```

図 3 myhttp 関数のプロトタイプ宣言

5 実験と考察

本章では，本研究で実装した通信ライブラリである myhttp 関数を使用し URL リクエストを行い，サーバとの通信切断時の送信データが再接続後に再送されることを確認する．

5.1 実験環境

Raspberry Pi 3 Model B をクライアント，Raspberry Pi 4 Model B をブローカとし，スイッチングハブを介して 1000Base-T ネットワークで接続してエッジネットワークとした．クラウドサーバとして AWS 上の t2.micro インスタンス (OS: Ubuntu 20.04 LTS) を用いて nginx 1.18.0 を起動した．

5.2 実験内容

nginx 上で http リクエストの GET 送信で受け取ったデータを mysql データベースに保存し，データベースに保存されている全てのデータを返す php プログラムを作成した．URL リクエストで n 回目に数値データ $10+n$ を送信する．数値データを 10 回送信し，5 回目を送信する前後で通信を切断し，提案する通信ライブラリと従来の方式との動作の違いを確認する．

5.3 実験結果

本研究で提案する通信ライブラリを使用して行った結果を表 1 に，従来の方式で行った結果を表 2 に示す．

表 1 提案する通信ライブラリを使用時のデータベース内のデータ

回数	1	2	3	4	5	6	7	8	9	10
データ	11	12	13	14	15	16	17	18	19	20

表 2 従来の通信方式を使用時のデータベース内のデータ

回数	1	2	3	4	5	6	7	8	9
データ	11	12	13	14	16	17	18	19	20

5.4 考察

表 1，表 2 より，従来の通信方式と比較して，提案する通信ライブラリはデータを 1 つ多く送信していることがわかる．これは 5 回目の通信時の通信切断時でのデータであり，提案する通信ライブラリは通信切断時にもブローカに送信できなかったデータを一時的に保存し，データを損失することなく再送信できている．

6 おわりに

本研究では，エッジネットワークとクラウド間の通信が切断されている場合でも，エッジネットワーク上のブローカを利用して送信データの損失を抑える通信ライブラリを提案した．実際に通信切断時には Kafka ブローカにデータを送る通信ライブラリを作成し，データの損失無く通信切断に対応できているかを実験した．実験の結果，本研究での通信ライブラリは，使用していないものと比較すると通信切断時にも適切に Kafka ブローカにデータを送信することができ，データの損失を防げていることを確認した．

今後の課題としては，通信切断時に新規データが送信されるような状況にも対応できるように実装を改善することが挙げられる．現在の実装ではマルチスレッドクライアントで通信切断時にも新しいデータを送信する場合にブローカに保持する通信データの順序が保存されない．このような場合にも正常に動作するように機能追加する必要がある．

参考文献

[1] 福田　敦, 吉田　景, 早川　智一：”不意の通信切断を考慮した WebSocket エミュレーションフレームワークの提案” 情報処理学会第 82 回全国大会, Vol1, pp.139-140,(2020,2)