

アクセス制御を可能にする位置情報通信プロトコルに関する研究 A Study on a Communication Protocol of Location Information for an Access Control

谷川 広和[†] 長坂 康史[†]
Hirokazu Tanigawa Yasushi Nagasaka

1. まえがき

近年、GPS(Global Positioning System)を利用することができるネットワーク端末が普及しており、今後もこのような端末は増えていくと考えられる。現在は、これらの端末向けに、ネットワークを介してナビゲーションサービス、位置情報を利用したコミュニティサービス、現実の映像にエアタグなどのデジタル情報を付与する拡張現実など様々なサービスが提供されている。また、行動履歴を利用したシステムなど^{[1][2]}の研究開発もされており、ネットワークと位置情報を組み合わせて利用するサービスは、欠かせない物となりつつある。

しかし、これらのサービスでは、サービスによって位置情報の取扱い方が違うため、汎用性がなく、システムやサービスによって、取り扱い方を一から考える必要がある。また、既存のサービスでは、ユーザーの居る場所の情報は活用されず、すべてのユーザーに同じ情報を提供している。そのため、その場所に必要な情報がその他の情報によって埋もれてしまうなどの課題がある。

そこで、本研究では、ネットワーク通信時における位置情報データの送受信方法の統一化を図り、ユーザーの位置によるネットワーク上のコンテンツやアクセスの制御を行うことが可能な位置情報通信プロトコルの研究を行う。

2. 提案プロトコル

クライアントが GPS を利用して取得した位置情報を、位置情報通信プロトコルとして用意したデータ単位に分割し、クライアントユーザーのメッセージとともにサーバーにデータ送信を行う。サーバーは位置情報通信プロトコルとして送られてきたデータをユーザーメッセージとユーザー位置情報に分割し、サーバーユーザーに伝達する。

提案プロトコルには、位置情報に必要な座標情報だけでなく、時間、日付、西暦情報を付加するとともに、位置情報通信プロトコルを判定するためのデータも付加する。これらの情報を付加した位置情報通信プロトコルのサイズは、20 Byte となる。図 1 にそれぞれの格納データを示す。位置データ取得日時を示すために、秒、分、時、日、月を付加する。また、位置情報通信プロトコルを用いた通信を行っているかどうかを判断するための判定用データも付加する。最後にユーザーの位置情報である座標データを付加する。

位置情報通信プロトコルの送信用関数を `send_pos`、受信用関数を `recv_pos` とする。これらの関数の引数や戻り値は、通常の `send` 関数や `recv` 関数と同じである。そのため、ユーザーは今までの `send`, `recv` を `send_pos`, `recv_pos` に変更するだけで、位置情報データの送受信が可能となる。

`send_pos` 関数は、ユーザーメッセージに位置情報プロトコルを付加するため、`send_pos` 関数で送信したデータを通常の `recv` 関数で受信することができない。



図 1 位置情報通信プロトコルフッター

2.1 send_pos

位置情報通信プロトコルを実装することで、クライアントは位置情報の送信を意識することなくデータの送信を行う。

`send_pos` 関数は、本来ユーザーが送信したいデータに位置情報通信プロトコル用のデータとして、位置情報を付加し、サーバーにデータを送信する。

2.2 recv_pos

送信側が位置情報通信プロトコルを利用した場合、データ部に挿入されているメッセージと位置情報通信プロトコルを分割しなくてはならない。しかし、受信側は、位置情報通信プロトコルを使用した通信と通常通信のパケットの両方を受け取る可能性がある。そこで、位置情報通信プロトコルでは、`send_pos` 側で付加された判定用のデータを受信側の `recv_pos` で読み取り判断する。判定用データがない場合は通常通信と判断する。

3. 位置情報通信プロトコル実装システム

クライアント端末は、GPS を用いて取得した位置データを、端末のネットワークを利用し、WEB ページのソースである `index.html` を要求する。サーバーは位置情報通信プロトコルを利用し、ユーザーの位置情報に適合する `index_*.html` を返す。

図 3 にその例を示す。この例では、前提条件として、クライアントの場所はそれぞれ異なる場所としている。クライアント 1, 2, 3 も要求は同じ `index.html` であるが、返信されるデータはそれぞれ、`index_1.html`、返信不可、`index_2.html` を返信し、クライアント 1、クライアント 3 はそれぞれ異なるページを表示するが、クライアント 2 は WEB ページを閲覧することができない。

このように、本論文で提案している位置情報通信プロトコルを利用することで、クライアント端末のネットワーク

[†] 広島工業大学大学院工学系研究科
Graduate School of Engineering, Hiroshima Institute of Technology

上での位置によるコンテンツ制御およびアクセス制御を行うことが可能となる。

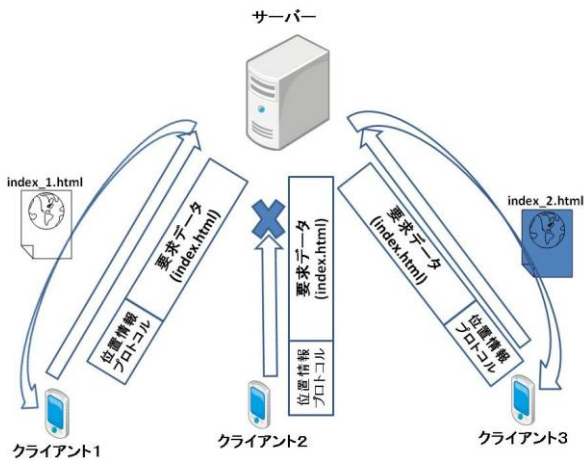


図3 位置情報通信プロトコル実装システム全体図

4. 性能評価

性能評価として、位置情報通信プロトコルを実装した際のオーバーヘッドの測定を行う。具体的には、ネットワークを介して接続された同スペックの2台のLinuxマシンを利用して、クライアントがパケットを送信してから受信するまでの時間を2分割したものを送信時間として計測した。図4及び図5にその結果を示す。図では、位置情報通信プロトコルを実装した通信をPOS_TCP、POS_UDPとし、通常の通信をNormal_TCP、Normal_UDPとする。

表1 マシンスペック

Machine	
OS	Linux(2.6.18-238.12.1.el5)
CPU	Intel(R) Celeron™ 1200MHz
Memory	512M

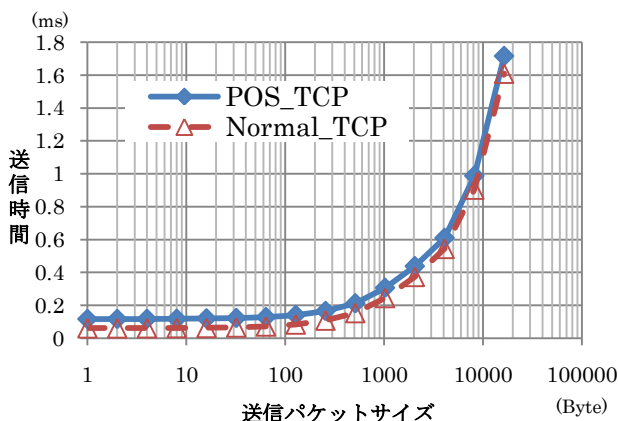


図4 TCP通信における送信速度の比較

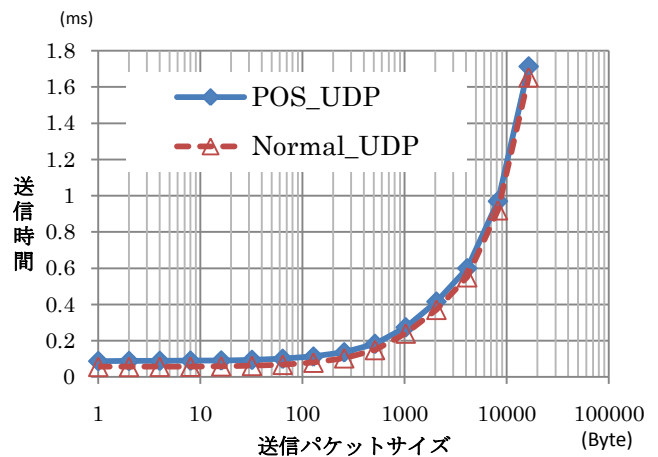


図5 UDP通信における送信速度の比較

図4、図5からわかるように、位置情報通信プロトコルを利用した通信は、通常通信よりも時間がかかっており、その通信時間差はTCP、UDP共に0.1ms以下である。

5. 考察

位置情報通信プロトコルを利用したTCP、UDP通信の場合、通常のTCP、UDP通信より時間がかかっている。これは、位置情報通信プロトコルを追加したことによる送信データサイズの増加と、受信後の処理が原因である。また、位置情報通信プロトコルによる通信と通常通信の差は、送信開始時からほぼ一定であり、送信時間の差もUDP、TCP共に0.1ms以下であった。そのため、位置情報通信プロトコルの通信に大きな問題もなく、位置情報通信プロトコルを付加した通信を行ったとしても、実践で使える範囲であると言える。

6. おわりに

本研究では、ネットワークを介して位置情報を送信する際の規約として、位置情報通信プロトコルの提案を行った。位置情報通信プロトコルを実装したシステムでは、HTTP通信におけるコンテンツ制御及びアクセス制御について述べた。また、評価として、Linuxを実装したマシンを用いて、提案プロトコルのオーバーヘッドの計測を行った。結果は、位置情報通信プロトコルを利用した場合の通信速度と、通常の通信の速度の差は0.1ms以下であるため、実用可能範囲と言える。

今後の課題として、位置情報を利用した既存サービスへ対応させるために、位置情報通信プロトコルのオプションの追加及びAPIを用意するなど、汎用性の高いフレームワークを検討する必要がある。

参考文献

- [1] 田島 孝治, 安藤 公彦, 大島 浩太, 寺田 松昭, “行動履歴に基づく予測型情報提示システム「水晶珠」の試作”, 電子情報通信学会論文誌B, Vol.J92-B, No.7 (2009).
- [2] 竹内 雄一郎, 杉本 雅則, “位置情報履歴を利用したユーザアダプティブな街案内システム”, 電子情報通信学会論文誌D, Vol.J90-D, No.11 (2007).