

## デバイス移動に対しても サービスが継続可能な遠隔デバイスアクセス機構

尾崎 亮太<sup>†</sup> 日高 宗一郎<sup>††,†</sup>  
児玉 和也<sup>††,†</sup> 丸山 勝巳<sup>††,†</sup>

計算機に接続されたデバイスにネットワーク上の別の計算機からアクセス可能とする技術は、ネットワーク構成や計算機やデバイスの接続方式に左右されることなく遠隔デバイスアクセスを実行し、環境の変化の影響を可能な限り隠蔽することが望まれる。本論文では、このような要求に応える遠隔デバイスアクセス機構について報告する。実現したシステムは、OS が提供するデバイスアクセス API による遠隔デバイスアクセス、ファイアウォールによりアクセス制限があるサブネットワーク間の遠隔デバイスアクセス、デバイス接続先計算機の変化に対してサービス継続が可能などの特徴を持つ。またデバイス特性を基にした評価実験により、NextD が遅延への要求が高いデバイスの性能要件を満たしていることがわかった。

### Design and Implementation of Remote Device Access Facility to Support Device Migration

RYOTA OZAKI,<sup>†</sup> SOICHIRO HIDAKA,<sup>††,†</sup> KAZUYA KODAMA<sup>††,†</sup>  
and KATSUMI MARUYAMA<sup>††,†</sup>

Technologies to access remote devices from mobile computers over network are drawing attentions. Nowadays, growth of wireless and hot-pluggable devices causes a new problem: dynamic change of network configuration such as and device connective configurations such as a device migration. We propose a system called "NextD" which makes it easy to cope with such problems while keeping conventional APIs intact. NextD allows applications to access remote devices in exactly the same way as they access local devices, through device file interface. NextD supports a host migration and a device migration enable application to continue accessing to remote devices. This paper describes design and implementation of NextD. We have implemented a prototype system on Linux OS. Experimental results show that required features are naturally supported and the overhead of the proposed system is small enough for actual use.

#### 1. はじめに

無線 LAN 内蔵計算機や無線接続デバイスの普及に伴い、ネットワーク経由で他の計算機のデバイスにアクセスしたり、移動時の接続切替を意識することなく計算機やデバイスを持ち歩きながら使いたいという要求が高まっている。特にモバイル環境では入出力デバイスが制限されているので、オフィス内では備え付け計算機の使いやすいデバイスを使いたい、あるいは使い慣れたデバイスを持ち歩いて使いたいという要求

は高い。

本論文では、このような要求に応える遠隔デバイスアクセス機構 NextD ( Network EXTented Device management system <sup>1)~3)</sup> ) について報告する。NextD の概要は以下の通りである ( a ) デバイス移動: デバイスの接続先切替わりが起きても、遠隔デバイスアクセスサービスが継続される ( b ) デバイスファイルによる遠隔デバイスアクセス: 遠隔デバイスをローカルのデバイスファイルとして見せ、OS が提供する既存 API によるアクセスを可能とする。これにより既存アプリケーションでも遠隔デバイスアクセスが可能である ( c ) サブネットワーク越え遠隔デバイスアクセス: ファイアウォールによる制限があるネットワークでも、遠隔デバイスアクセスを可能とする。なお開発の容易化と拡張性を高めるため、機能の大部分はユーザレベ

<sup>†</sup> 総合研究大学院大学 情報学専攻, Department of Informatics, The Graduate University for Advanced Studies.

<sup>††</sup> 情報・システム研究機構 国立情報学研究所, National Institute of Informatics, Research Organization of Information and Systems.

ルプログラムとして実現した。また本論文では、評価実験結果について報告する。デバイス特性を基にした評価実験により、NextD が遅延への要求が高いデバイスの性能要件を満たしていることがわかった。

以降、本論文は次のように構成される。2章で遠隔デバイスアクセス技術の利用法と要件、3, 4章では実現方法の詳細、5章で評価実験結果、6章で関連研究との比較評価について報告し、最後に7章で本論文をまとめる。

## 2. 遠隔デバイスアクセス技術の利用法と要件

### 2.1 利用法

旧来の遠隔デバイスアクセス技術は、固定計算機を対象としていた<sup>4)~7)</sup>。計算機やデバイスの存在は既知で変わらないため、遠隔デバイスアクセス機構には単に遠隔デバイスアクセスするための機能しか必要ではなかった。ところが計算機やデバイスの小型軽量化や無線技術の発達により、遠隔デバイスアクセスには新たな要求が生まれた。ユーザは計算機や無線デバイスを持ち歩き、移動先にある計算機やデバイスと組み合わせ利用することが考えられる。次節では、このような環境の中で生まれる遠隔デバイスアクセス技術の要件を挙げる。

### 2.2 利用環境

また本研究では、主に入出力デバイス(マウス、キーボード、スピーカなど)を遠隔デバイスアクセスの対象とする。計算機との接続方式としてはUSBなどの有線接続とBluetooth<sup>8)</sup>などの無線接続のいずれも対象とする。以下本論文では、特に断りがない限り、デバイスはここで述べた入出力デバイスを指す。

### 2.3 要件

- デバイスの持ち歩き  
モバイル計算機や無線デバイスをユーザが持ち運びながら遠隔デバイスアクセスを行なうことが考えられるが(例えばヘッドセットで本部と連絡をとりながら巡回する警備員)、デバイスの接続先計算機が切り替わるたびに手を煩わせるシステムを、ユーザは利用したいと思うだろうか?
- 遠隔デバイスアクセス  
もしユーザが貧弱なデバイスしか具備しない計算機を利用しているならば、他の計算機のリッチなデバイスを利用したいと思うかもしれない。その際、特定のアプリケーションしか利用できなかったり、利用するのに手間がかかることを、ユーザは望まないだろう。
- 別サブネットワークのデバイスの利用

遠隔デバイスアクセスサービスが提供されているが、それはサブネットワーク内の計算機に限定されるとしたらユーザはどう思うだろうか?社内で別の部署から自分のデスクの計算機にはアクセスできないという制限は喜ばしいものではない。またファイアウォールによって、遠隔デバイスサービスが利用できないことは、ユーザにフラストレーションを与えるだろう。

無線技術の普及し、利用方法が多様化するなか、遠隔デバイスアクセス技術には、以上のような問題を解決する機能が必要となる。

## 3. 遠隔デバイスアクセス機構 NextD

本章では遠隔デバイスアクセス機構 NextD の基本的な以下の機能について述べる。NextD が提供するサービスを、本論文では NextD サービスと呼ぶ。

- 計算機とデバイスの発見
- OS の既存 API による遠隔デバイスアクセス
- デバイスアクセス実行方法
- サブネットワークを越える遠隔デバイスアクセス

### 3.1 計算機とデバイスの発見

NextD は、遠隔デバイスアクセスサービスを提供する他の計算機を自動的に見つけサービスグループ(以下 NextD サービスグループ)を形成する。NextD サービスグループ内で、デバイスの存在が周知され、アプリケーションはデバイスにアクセスすることができる。NextD サービスグループへの参加は、NextD 起動時と NextD が動く計算機がネットワーク接続した際に、UDP ブロードキャストで参加を表明することで行なわれる。NextD サービスグループ内では、デバイス情報を共有し、デバイスの着脱イベントを随時通知し合う。上記処理は NextD で自動的に行なわれ、アプリケーションやユーザはそれを意識する必要はない。NextD 間で交換される UDP メッセージを、本研究では NextD 制御メッセージと呼ぶ。

サブネットワーク間の NextD サービスグループ形成方法については、3.4 で述べる。

### 3.2 OS の既存 API による遠隔デバイスアクセス

NextD は、遠隔デバイスをローカルのデバイスファイルとしてアプリケーションに提供する。したがって既存 API で遠隔デバイスにアクセスできるので、特別な機能を持たないアプリケーションであっても、遠隔デバイスにアクセスすることができる。

### 3.3 デバイスアクセス実行方法

遠隔デバイスに対応するデバイスファイルへの操作( `open()`, `close()`, `read()`, `write()`, `ioctl()` )を、遠隔計

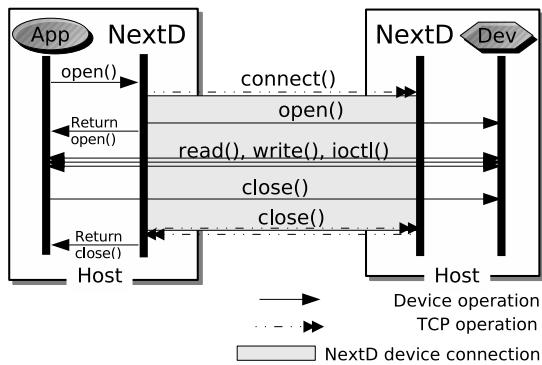


図 1 遠隔デバイスアクセスと NextD デバイスコネクション  
Fig. 1 Remote device access and NextD device connection

算機上で NextD が代わりに行なうことで、遠隔デバイスアクセスは実施される(図 1)。アプリケーションがデバイスファイルを open すると、NextD は遠隔の NextD との間で TCP コネクションを張り、以降の read(), write(), ioctl() などの操作はそのコネクションで行なわれる。コネクションはアプリケーションがデバイスファイルを close した際に閉じられる。以降本論文ではこのコネクションを NextD デバイスコネクションと呼ぶ。

### 3.4 サブネットワークを越える遠隔デバイスアクセス

#### 3.4.1 機能要件

遠隔デバイスアクセスサービスは、ユーザがネットワーク環境を意識することなく利用できることが望ましい。サブネットワークを越えた遠隔デバイスアクセスサービスを提供するためには、別サブネットワークの NextD を発見し、NextD サービスグループを拡大するためのメッセージ中継機構が必要となる。

ところで、近年のセキュリティ意識の高まりにより、ファイアウォールが設置され、特定の計算機に特定のポートに対する通信のみを許す環境が一般的である。中継機構の実現においては、そのような制限されたネットワークを考慮する必要がある。

#### 3.4.2 Mediator

Mediator は、ネットワークを跨いだ NextD サービスを実現するためのプログラムである。ネットワーク管理者が許可した特定の TCP ポートを介して外部サブネットワークとやりとりをする。あらかじめ別サブネットワークの Mediator とコネクションを張り、自サブネットワークの NextD 制御メッセージやデバイスデータを別サブネットワークへ転送する。こうしてサブネットワークをまたいだ NextD サービスグループが形成される。

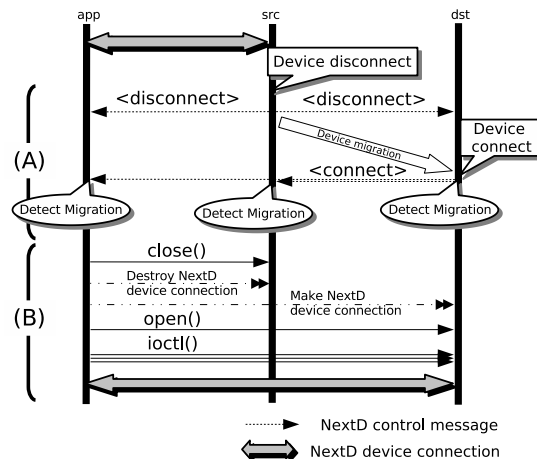


図 2 デバイス移動時の処理  
Fig. 2 NextD processing on device migration

以下は Mediator が NextD に提供する主要機能である。

- NextD 制御メッセージ転送  
Mediator は、NextD 制御メッセージを NextD サービスグループを形成する別サブネットワークの Mediator へ転送する。これによりサブネットワーク間の計算機の存在やデバイス情報を共有することができる。
- NextD デバイスコネクション中継  
直接 TCP コネクションを張ることができない別々のサブネットワークの NextD 間の仲立ちをする。Mediator は NextD とコネクションを張ることで別の NextD と通信を行なう。

### 3.5 実装

本研究では Linux OS 上に、ユーザレベルでファイルシステムを実装するためのフレームワークである FUSE を用い、NextD を実現した。そのため大部分の機能はユーザレベルプログラムとして実現されている。

## 4. デバイス移動

デバイス移動とは、ユーザの無線デバイス持ち歩きなどにより、デバイスの接続先計算機が切り替わることをいう。本章では、デバイス移動が起きてもアプリケーションが遠隔デバイスアクセスを継続することを可能とする機能について、処理手順に沿って説明する(図 2)。

<http://fuse.sourceforge.net/>  
カーネルレベルの変更は、ioctl() 実現のため FUSE へ追加した 200 行程度である。

#### 4.1 デバイス移動検出

NextD は、デバイスの接続・切り離しイベントは Linux OS のデバイス管理デーモン hald から受け取る。

##### (1) デバイス切り離し

NextD は、hald からデバイス切り離しイベントを受け取ると、デバイス切り離しメッセージを NextD サービスグループへマルチキャストする。メッセージにはデバイス同定に用いるシリアル番号などが含まれる。シリアル番号を持たないデバイスの場合は、デバイスはデバイス種別で同定を行なう。この方法は、NextD サービスグループ内に同種のデバイスが一つしか存在しない状況でのみ用いることができる。

NextD はデバイス切り離しメッセージを受け取ってから、一定時間（ユーザが設定可能）以内にデバイス接続メッセージを受け取ることで、デバイス移動が起きたことを知る。

##### (2) デバイス接続

NextD は、hald からイベントを受け取るとデバイス接続メッセージを NextD サービスグループへマルチキャストする。これで図 2 のようにすべての NextD でデバイス移動が検出され、NextD デバイスコネクション張替えの準備が整う。

#### 4.2 コネクション張替えとデバイス状態の復帰

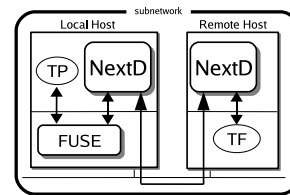
##### (1) NextD デバイスコネクション張替え

アプリケーションの動いている計算機の NextD は、デバイス移動を検出すると移動先の NextD へコネクションを切替える。同時に、移動元計算機のデバイスファイルを close し、移動先計算機のデバイスファイルを open する。

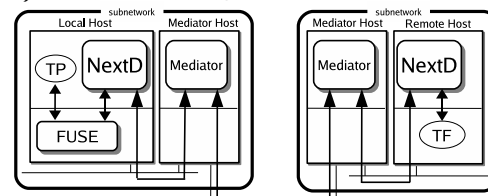
##### (2) デバイス状態の保存・復帰

デバイス移動が起きてデバイスが新たな計算機に再接続されたときには、デバイス状態がリセットされる。そのためデバイス状態を切り離し前の状態に戻さなければならない。

アプリケーションは ioctl() を用いてデバイス固有の設定を行なう。本研究ではアプリケーションが発行する ioctl() を保存し、デバイスが再接続した後に保存した ioctl() を再発行することでデバイス状態を復帰している。



(A) サブネットワーク内



(B) サブネットワーク間

TP - Test Program  
TF - Target File

図 3 計算機・ソフトウェア構成

Fig.3 Computer and Software Configuration

#### 4.3 サブネットワーク間デバイス移動

サブネットワークをまたいだデバイス移動時の処理も、デバイス切り離し・接続メッセージの転送や NextD デバイスコネクションの確立に Mediator を介する点をのぞいて上記処理手順と同じである。

#### 5. 評価実験

本章では、デバイスごとに要求される性能を明らかにし、遠隔デバイスへのアクセス遅延を計測する実験およびデバイス移動時のサービス再開までの処理時間を計測する実験の結果について報告する。

##### 5.1 デバイス性能要件

本論文では、入出力デバイスの代表例として、USB デバイスクラスという HID ( Human Input Device ) と Audio を取り上げる。

- HID: マウスであれば、ユーザがマウスを動かしたら、即座にマウスカーソルが動くという即時フィードバックが求められる。反応時間が 100 ms 以下であることが、人間が即座に応答していると感じる一つの基準である<sup>11)</sup>。Linux OS の場合、データサイズはマウスで 3 bytes ( イベント形式ではデータサイズは 16 ~ 48 bytes ) である。つまり 64 bytes 程度のデータを 100 ms 以内に処理できる性能が HID に対する性能要件である。
- Audio: スピーカに対する音楽データ出力であれば、音が途切れないこと求められる。MP3 などの音楽データのビットレートは 56 ~ 160 Kbps が一

<http://www.freedesktop.org/Software/hal>  
Vendor ID と Product ID で決まる。

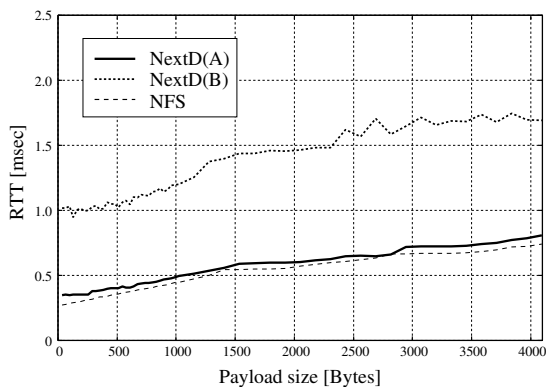


図 4 データ書き込み遅延

Fig. 4 Latency of write processing

一般的であるが、例えば音楽再生プログラム mpg123 は、160 Kbps のデータを 4 Kbytes ずつ、5 回/秒程度の頻度で出力する。つまり 4 Kbytes のデータを 200 ms (1,000 ms/5 回) 以内に処理できることが、Audio に対する性能要件である。

以下の実験は、この要求を満たす性能を得られているかどうかを確認することを目的とする。

## 5.2 評価実験(遅延)

### 5.2.1 実験環境

NextD と Mediator におけるデータ書き込み遅延を計測するため、2 つの実験環境で実験を行なった(図 3)。実験(A)は Fast Ethernet で接続した 2 台の計算機(ともに CPU 1.0 GHz)上で NextD が動かし、一方の計算機から他方の計算機のデバイスファイルに書き込みを行なう。実験(B)は Gigabit Ethernet で相互接続された 2 台の計算機(ともに CPU 2.6 GHz)上で Mediator を動かし、実験(A)の NextD を中継する構成である。

計算機間の通信にはすべて TCP が用いられている。サイズが小さいデータを即座に送出するために、TCP の Nagle アルゴリズム を無効にした。また比較に NFS (Network File System) を用いた。

### 5.2.2 結果と考察

図 4 は各ペイロードサイズ(64 bytes から 4 Kbytes)に対するデータ書き込み遅延である。

実験(A)では、ペイロードサイズ 64 bytes で 0.35 ms と HID の許容遅延とされる 100 ms に比べ十分

小さい。また 4 Kbytes で 0.80 ms と、Audio に求められる 200 ms に比べ十分小さい。NextD は図 3 のように書き込みデータがカーネルから一度ユーザに渡されるが、その処理時間は小さい。ペイロードサイズ 64 bytes と 4 Kbytes でも NextD の処理時間は 0.12 ms で残りは通信遅延である。また NextD は NFS と比べ遜色ない性能を得ている。

実験(B)では、データ書き込み遅延はペイロードサイズ 64 bytes で 1.0 ms、4 Kbytes で 1.7 ms であり、Mediator を介してもデバイスの性能要件を満たしている。

以上の結果から NextD が HID や Audio など遅延への要求が高いデバイスの性能要件を満たしていることがわかった。

## 5.3 評価実験(デバイス移動)

USB スイッチを用い、2 台の計算機で 1 つの USB デバイスを切り替えデバイス移動を模擬する実験を行ない、デバイス移動からサービス再開までの処理時間を計測した。

デバイス切り離し(図 2 における Device disconnect)から音声が再び USB スピーカから流れ出すまでに約 5.0 秒の時間を要した(デバイス切り離しから再接続までの時間は含まれない。)この時間は決して小さなものではないが、NextD の処理時間は実は大きくない。処理時間の内訳は、Linux がデバイス装着イベントを検知して、hotplug (Linux の着脱イベント時に最初に呼び出されるユーザレベルプログラム)に通知するまでに約 0.7 秒、そこから hotplug や udev などのユーザレベルプログラムの処理を行なったのち NextD に着脱イベントが通知されるまでに約 3.0 秒、NextD の処理に約 1.0 秒となっている。そのため hotplug の次に NextD を呼び出す改良を加えることで、3.0 秒程度処理時間を短縮することが可能である。

またデバイス状態の復帰機能の効果を調べる実験を行なった。デバイス状態の復帰機能を有効にした実験では音声が正しく再生されたが、デバイス状態の復帰機能を止めた実験では、デバイス移動後に雑音が USB スピーカから流れてきた。これは mpg123 が初期化時に行なうデバイスの設定 をしていないためである。これによりデバイス状態の復帰機能が有効に働くこと、またデバイス状態の復帰はデバイス移動に対応するためには必須の機能であることがわかった。

RFC-896: Congestion Control in IP/TCP Internet-networks.

NextD は、遠隔計算機での write() 処理の戻り値を得るためローカル計算機で待ち合わせしており、データ書き込み処理は 2 回の通信を含む。

ioctl() の種類でいうと SNDCTL\_DSP\_SETFMT、SNDCTL\_DSP\_STEREO、SNDCTL\_DSP\_SPEED である。

## 6. 関連研究

Plan9<sup>4)</sup> や RFS<sup>5)</sup> では、OS やファイルシステムのプリミティブとして遠隔デバイスアクセスのための機能を実現している。いずれのシステムも遠隔デバイスのための API は OS の標準的なファイル操作インタフェースである。しかしいずれもデバイス発見や準備処理(デバイスファイルのマウント)はユーザが行なわなければならない。また計算機移動やデバイス移動に対応しサービスを継続することはできない。

PlanB<sup>10)</sup> は分散 OS Plan9 へ環境の動的変化に適応できる機能拡張を施した OS である。ファイルと資源の結合を実行時に動的解決することで、環境の変化に対応することができる。しかし NextD のようにデバイス移動には対応していない。

## 7. おわりに

遠隔デバイスアクセス技術は、ネットワーク構成や計算機やデバイスの接続方式に左右されることなく遠隔デバイスアクセスを実行し、環境の変化の影響を可能な限り隠蔽することが望まれる。

本研究では、このような要求に応える遠隔デバイスアクセス機構 NextD を Linux OS 上に実現した。NextD は OS が提供するデバイスアクセス API による遠隔デバイスアクセス、ファイアウォールによりアクセス制限があるサブネットワーク間の遠隔デバイスアクセス、デバイス接続先計算機の変化に対応してもサービス継続を可能とするなどの機能を持つ。NextD を使うことにより、ネットワーク経由で他の計算機のデバイスにアクセスしたり、移動時の接続切替を意識することなく計算機やデバイスを持ち歩きながら使うといった、モバイル環境での新たなユーザ体験を得られる。

またデバイス特性を基にした評価実験を行ない、NextD が HID や Audio など遅延への要求が高いデバイスの性能要件を満たしていることを示した。なお NextD は、C 言語コードが 2.3 K 行、Ruby スクリプト言語コードが 1.4 K 行規模のプログラムである。

今後の展開としては、より詳細な評価実験や、計算機移動後の再接続先に、可能であれば移動元サブネットワークの Mediator を選ぶといった最適化処理の実現などが考えられる。

謝辞 本研究を進めるにあたり、国立情報学研究所橋爪宏達教授、佐藤一郎助教授から有益な助言を頂きました。ここに感謝の意を表します。

## 参考文献

- 1) 尾崎亮太, 丸山勝巳, 日高宗一郎, 児玉和也: ネットワーク上に動的に分散する多数のデバイスを制御する基盤ソフトウェアの検討, 情処学 OS 研報 (SWoPP2004), 2004-OS-97, pp.81-88 (2004).
- 2) 尾崎亮太, 日高宗一郎, 児玉和也, 丸山勝巳: ネットワークの動的変化に対応する遠隔デバイス管理機構, 情処学 MBL/UBI 研報, 2005-MBL-32/2005-UBI7-8, pp.53-59 (2005).
- 3) Ozaki, R., Hidaka, S., Kodama, K. and Maruyama, K.: Accessing Remote Devices by Using Conventional Interface in Mobile Computing Environment, Proc. *the 8th International Workshop on Mobility in Databases and Distributed Systems (MDDS2005)*, pp.86-87 (2005).
- 4) Pike, R., Presotto, D., Dorward, S., Flandra, B., Thompson, K., Trickey, H. and Winterbottom, P.: Plan 9 from Bell Labs, *Computing Systems*, Vol. 8, No. 3, pp.221-254 (1995).
- 5) Rifkin, A.P., Forbes, M.P., Hamilton, R.L., Sabrio, M., Shah, S. and Yueh, K.: RFS Architectural Overview, Proc. *USENIX Conference*, pp.248-259 (1989).
- 6) Welch, B.B. and Ousterhout, J.K.: Pseudo Devices: User-Level Extensions to the Sprite File System, Proc. *the 1988 Summer USENIX Conference*, pp.37-49 (1988).
- 7) Hirofuchi, T., Kawai, E., Fujikawa, K. and Sunahara, H.: USB/IP - A Peripheral Bus Extension for Device Sharing over IP Network, Proc. *the USENIX 2005 Annual Technical Conference, FREENIX Track*, pp.47-60 (2005).
- 8) Bluetooth SIG. <http://www.bluetooth.com/>
- 9) Zandy, V.C. and Miller, B.P.: Reliable Network Connections, Proc. *the 8th annual international conference on Mobile computing and networking*, pp.95-106 (2002).
- 10) Ballesteros, F.J., Castro, E.M., Muzquiz, G.G. and Algara, K.L.: A New Network Abstraction for Mobile and Ubiquitous Computing Environments in the Plan B Operating System, Proc. *Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, pp.112-121 (2004).
- 11) Miller, R.B.: Response time in man-computer conversational transactions, Proc. *AFIPS Fall Joint Computer Conference*, Vol.33, pp.267-277 (1968).
- 12) Lightweight Directory Access Protocol (v3), RFC 2251 (1997).