

情報コンセントのための ハブポート対応表調査生成ツール

齊 藤 明 紀^{†1}

持ち込み PC 用自習室などでの多数並んだ情報コンセントの配線を効率よく調査してポート対応表を生成するツールを試作した。L2 結線状況を調査する手法はすでに多く提案されているが、情報コンセントの場合は個々のポートに機器を接続しては L2 接続探求ソフトを起動する手間が問題になる。電池動作の超小型 linux 機で情報コンセントから調査サーバに調査要求を送信する方式をとることで、作業員一人で調査作業を行うことができるようにした。

A tool to search and generate information outlet vs. hub port number table

AKINORI SAITOH ^{†1}

In this report, I propose a tool to generate a table that shows information outlet vs. hub port number correspondence. There are many method proposed to inspect Layer2 connection. But for information outlet, it is needed to reduce human resource needed to visit each outlet and invoke inspection tool. I designed a portable tool based on battery powered micro Linux machine and an inspection server. The portable tool involve an inspection process on the inspection server located somewhere over LAN. With this system only one person is needed to perform inspection.

1. はじめに

LAN 敷設から年月が経つと、情報コンセントの配線情報が後継担当者に引き継がれない、

紛失するといった事態が起こることがある。また、補修・改修・機器更新の際の配線変更の記録がなされなかったなどで、配線情報が実際と乖離してしまうことがある。特に経費節減のため、専門業者に発注せず管理者自ら工事するなどの際はケーブルのタグ付けを怠るなどして後日の混乱を招くことがある。本報告では、情報コンセントの各ポートがどのハブのどのポートに収容されているのかの対応表を効率よく作成するための調査ツールの試作について述べる。その際、調査作業を一人でも行えるように留意した。

2. 考察と方式設計

本研究で情報コンセントには管理上の名前がつけられているものと仮定する。これを情報コンセント識別子と呼ぶことにする。たとえば本学では「部屋番号-連番」である。教育研究棟(第4棟)5階の東端の部屋(1号室)は4501であり、その部屋の3番目の情報コンセントは「4501-03」となる。情報コンセント識別子はあらかじめ決まっているものでもよいし、情報コンセント敷設工事の際に決めた識別子も失われていた場合には新たに決めた識別子でもよい。

ネットワークの2つの機器間のL2/L3接続状況を調査する手法はすでに多く提案されている^{1),2)}。ただし調査のためには、情報コンセントに何らかの機器を接続しなければならない。本研究では多数の情報コンセントが設備された教室等を想定し、多数の情報コンセントを効率よく調査することを目標とする。

端末がハブのどのポートの下流にあるかの調査は、SNMPのBridgeMIB^{3),4)}等でハブのフォワーディングテーブル(FDB: Forwarding Data Base)を調べればよい。ただし、セキュリティ上の理由から、ハブは端末側からのSNMP問い合わせを許さなかったり、あるいはハブの管理モジュールをハブがサービスする情報コンセントとは別のVLANに置き、端末側からルーティングしないといった運用がしばしば見られる。情報コンセントの調査を行うためにハブの設定を一時的に変えるのは面倒である。そこで、FDBは情報コンセントに接続した機器が行うのではなく、ハブにアクセス可能なセグメントに接続した調査サーバが行うことにした。

全体構成の概念図を1に示す。LANは1つ以上のルータによっていくつかのブロードキャストドメインに分割されている。開発に当たって、調査する情報コンセント群がひとつのブロードキャストドメインに属することを前提とはしない。また、調査サーバが端末と同じブロードキャストドメインに置くことも前提としない。

本システムでは、作業員は調査端末を順次情報コンセントに接続し、調査サーバにipV4

^{†1} 鳥取環境大学環境情報学部情報システム学科
Tottori University of Environmental Studies

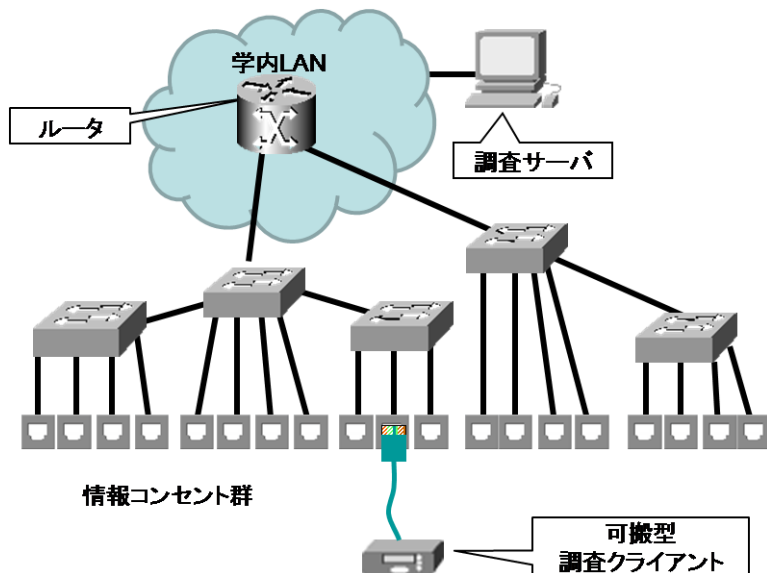


図 1 情報コンセントシステムの全体概念図

で通知を送り調査を行わせる。調査結果は調査サーバ側に蓄積し、あとでそこから表を生成する。

調査サーバは別セグメントに存在する場合もあるので、調査クライアントの MAC アドレスを知ることが出来ない。そこで、情報コンセント識別子と MAC アドレスはクライアントがサーバに通知することにする。

作業員、調査クライアント、調査サーバの動きは情報コンセント一カ所ごとに、おおむねつぎのようになる (図 2)。

- (1) 作業員は調査クライアントを情報コンセントに取り付け、情報コンセント識別子を入力、調査開始を指示。
- (2) 調査クライアントは NIC を稼働させ IP アドレス等を設定する。
- (3) 調査クライアントがサーバに情報コンセント識別子等を通知する。
- (4) 調査サーバはハブの FDB にアクセスしてポートを知る。
- (5) 調査サーバは調査結果 (ハブの IP アドレス、ポート番号、情報コンセント識別子) を記録に残す。

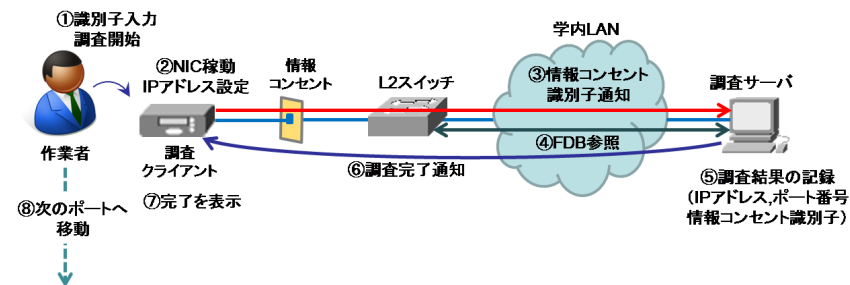


図 2 システム各部の動き

- (6) 調査サーバは調査クライアントに完了を通知。
- (7) 調査クライアントは調査完了を表示する。
- (8) 作業員は調査クライアントを情報コンセントから取り外し、次の席に移動する。

3. 試作と実装

本節では試作したシステムとその実装について述べる。

3.1 操作手順の策定

情報コンセント識別子を現場で手入力するのは煩雑である。そこで、情報コンセント識別子とその配置をあらかじめファイルに入力しておくことにする。

端末に対して上下左右矢印キーを押すと、現在位置の上下左右方向の隣接した情報コンセント識別子が表示され、そこで Enter キーを押すとサーバとの通信が行われるようにする。例えば図 3 のように 5 席の長机が 2 × 5 に並び、また全面の壁にも情報コンセントがある部屋を考える。ここで p01 p02 ...p05 p15 P14 ... p11 と折り返しながら移動して調査する場合には、

	Enter		Enter		Enter		Enter		Enter
	Enter		Enter		Enter		...		Enter

とキー操作すればよいことになる。

そのために、図 4 のように定義ファイルを書けばよい。碁盤の目で近似してコンセントの位置を表し、情報コンセント一列分の識別子を 1 行に、空白文字区切りで記述する。欠けている位置には マイナスを記入する。このような定義ファイルは、スクリプト言語や excel のオートフィル機能を使えば容易に生成できる。なお、後述するように調査端末の表示装置にキャラクタ LCD を使用した都合で、情報コンセント識別子に使用可能な文字は ASCII

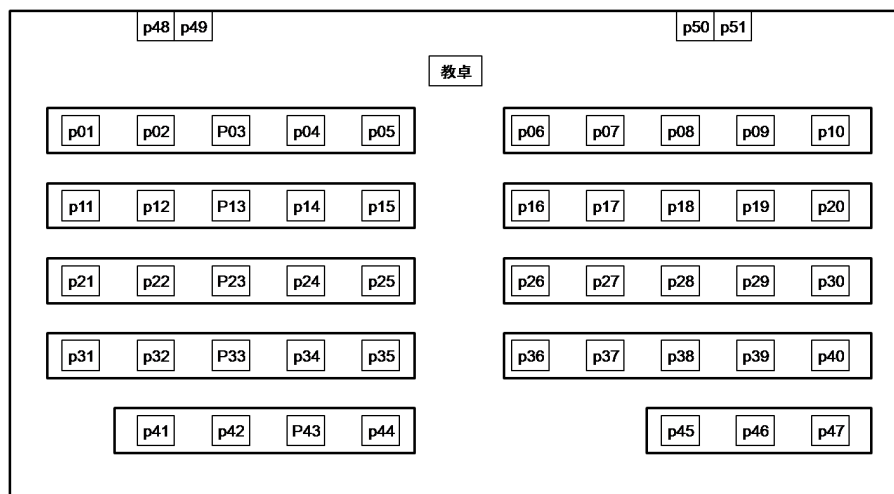


図 3 情報コンセント配置例

-	p48	p49	-	-	-	-	-	p50	51	-
p01	p02	p03	p04	p05	p06	p07	p08	p09	p10	
p11	p12	p13	p14	p15	p16	p17	p18	p19	p20	
p21	p22	p23	p24	p25	p26	p27	p28	p29	p30	
p31	p32	p33	p34	p35	p36	p37	p38	p39	p40	
-	p41	p42	p43	p44	-	-	p45	p46	p47	

図 4 配置定義ファイル例

文字に限られる。

このような調査端末を用いた作業手順は以下のようになる。

- (1) 調査したい情報コンセントに端末を接続する
- (2) 矢印キーで情報コンセント識別子を選択し, Enter を押す
- (3) 調査終了の表示を待って情報コンセントから端末を取り外す
- (4) 次の位置に移動

3.2 端末ハードウェア設計

調査クライアント端末 (以降端末) には超小型の Linux 機であるアットマークテクノ社製

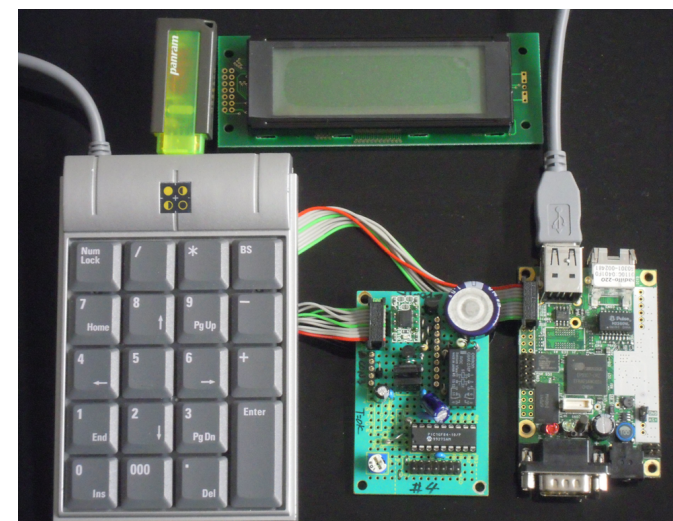


図 5 端末構成部品

Armadillo-220 を用いた。これにパソコン用 USB テンキーと小型液晶表示装置を接続する。これを単三型二次電池 eneloop で駆動する。データ供給は FAT32 ファイルシステムの USB メモリを用いる。

図 5 に端末の構成部品を示す。上段がキャラクタ液晶, 下段が左から, テンキーと USB メモリ, 液晶制御兼電源基板, Armadillo-220, 電池 BOX (単三 × 2, USB 出力) である。表示装置は秋月電子通商「シリアル制御液晶モジュールキット」*1 を用いた。LCD は 20 文字 × 4 行のキャラクタ液晶, 液晶制御は PIC16F84 を使い, armadillo との接続は調歩同期シリアルである。

3.3 端末ソフトウェア設計

本学環境ではすべての情報コンセントは DHCP サーバの管理下にあるので, IP アドレスやデフォルトルート等は DHCP で取得する方式で施策を行った。固定 IP 方式あるいは DHCP サーバが使用できない状況への対応は容易である。後述する MAC アドレス連番生成と同様に IP アドレスも連番生成すればよい。

*1 現在は販売終了

armadillo-220 はフレームバッファを持たない小型 linux 端末であり、コンソールはシリアルポートである。そのため USB テンキーに入力された文字をコンソール TTY で受け取ることができない。そこで linux input subsystem⁵⁾⁻⁷⁾ を用いることにする。armadillo 添付の linux カーネルでは、USB キーボードからの入力は/dev/input/event0 で受け取ることができる。USB テンキーもキーボードとして扱われる。キー押下イベントが発生するのを待ってキーコードを stdout に出力にするコマンド inkey を作成した。

inkey で得たキー押下情報をシェルスクリプトで処理して、現在着目している情報コンセント識別子を LCD に表示する。

Enter キーの押下を検出すると、以下の手順で調査を行う。

- (1) インターフェース eth0 を down 状態にする
- (2) インターフェース eth0 の MAC アドレスを変更する
- (3) DHCP で IP アドレス等ネットワーク接続パラメタを受け取り、設定する
- (4) 調査サーバに MAC アドレスと情報コンセント識別子を送る
- (5) 調査サーバの調査完了を待つ
- (6) DHCP で得た IP アドレスを返却する

ひとつの部屋の情報コンセントが異なるブロードキャストドメインに渡っていることも考えられる。その場合、新しく接続した情報コンセントからの Layer2 ブロードキャストは以前接続していたハブに届かず、MAC アドレスが FDB が更新 (抹消) されないことが考えられる。そこで念のため、MAC アドレスを接続ごとに変更することにした。MAC アドレス 6 オクテットの先頭オクテットの msb 側 2 ビットが、グローバル管理 (1)/ローカル管理ビット (1) とユニキャスト (0)/マルチキャスト (1) ビットである。そこで先頭バイトの下位 2 ビットが 10 であるような 6 バイト値を MAC アドレスとして設定すればよい。

試作では、設定ファイルで与えた値を先頭 4 オクテットに、0 から始まる 2 バイト値を末尾 2 オクテットに設定することにした。製作したプロトタイプでは、本学のグローバル ipv4 アドレスを先頭 4 オクテットに用いて、ca:38:4e:03:00:00 から始まる連番を用いた。

複数の端末を用いての並行調査作業を行う場合は、MAC アドレスが重複しないように設定しておく必要がある。

3.3.1 実装の詳細

組み込み linux 用ユーティリティ busybox⁸⁾ に含まれる DHCP クライアント udhcp を用いる。この dhcp クライアントプログラムは本来はデーモンとして使用するのだが、今回はフォアグラウンドプロセスとして起動する。そのため、dhcp や sshd ネットワーク系の daemon

```
echo tcpsend $server $ID $mac $name > /tmp/do_rep.sh
# start for $name
ifconfig down eth0
ifconfig eth0 hw ether $mac
udhcp -f -p /var/run/udhcp.eth0.pid -i eth0 \
-n -s /usr/share/udhcp/scan.script
```

図 6 端末での処理概要

```
read pid < /var/run/udhcp.eth0.pid
if [ -s /tmp/do_rep.sh ];then
sh /tmp/do_rep.sh
rm -f /tmp/do_rep.sh
fi
kill -USR2 $pid
kill -TERM $pid
```

図 7 udhcp スクリプトに追加した処理

の起動設定を解除しておく (/etc/rc.d の設定)。

調査端末で行う情報コンセント 1 か所ごとの処理を図 6 に示す。シェル変数 name には情報コンセント識別子が入っている。

udhcp は DHCP サーバとの通信のみを行い、実際にネットワークインターフェースに設定する作業は外部のシェルスクリプトが行う。そこで、udhcp が起動するシェルスクリプト内に、図 7 ような処理を追加した。

これにより、IP アドレスリソースが得られた状態でスクリプトが起動されると、ネットワークの諸設定を行った後に /tmp/do_rep.sh が起動される。/tmp/do_rep.sh の中に、調査サーバに情報を送信する処理が記述されている。続く二つのシグナル送信は、このシェルスクリプトが終了した後で、udhcp によって順に処理される。USR2 シグナル送信で udhcp はアドレスを DHCP サーバに返却する。続いて TERM シグナルで udhcp が終了する。

調査サーバへは、端末 ID (文字列)、MAC アドレス、端末を空白区切りのプレーンテキストで送信する (tcpsend コマンド)。送信終了後、tcpsend はサーバが TCP 接続を切断するのを待つ。調査サーバは、調査が完了してから TCP 接続を切断するように構成しておく

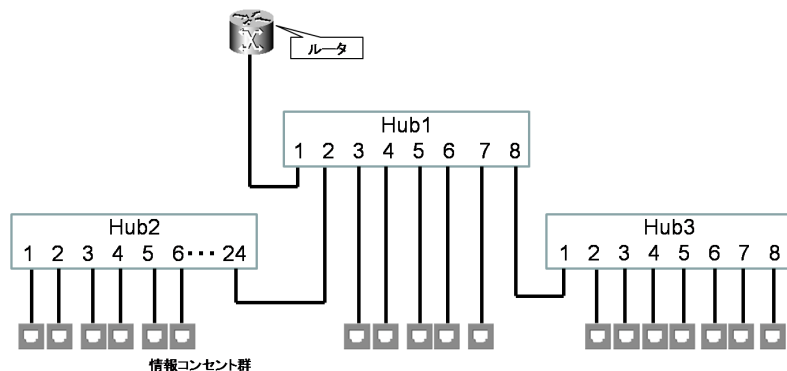


図 8 L2SW カスケード接続例

```
#hub community port-to-ignore
#port-to-ignore = uplink port or cascading port
hub2.example.ec.jp public 24
hub3.example.ec.jp public 1
hub1.example.ec.jp public 1 2 8
```

図 9 調査サーバ設定ファイル例

必要がある。

3.4 サーバソフトウェア設計

調査サーバは標準入力から空白区切りのテキストを 1 行受け取って、端末 ID,MAC アドレス、情報コンセント識別子に分離する。あらかじめ設定ファイルに書いておいた情報コンセント収容 L2 スイッチに順次 SNMP で問い合わせを行い、FDB を取得し、該当の MAC アドレスを捜す。また、調査作業が終るまでクライアントとの TCP 接続は保持する。

今回は実装の手間を省くため、L2SW のカスケードポートはあらかじめ調査して設定ファイルに書いておくことにした。

図 8 に例示するような構成の場合は、図 9 のような設定ファイルを書けばよい。カスケード接続関係にあるハブは下流のものから先に書く必要がある。

調査を行うプログラムは /usr/local/libexec/lansockrep であり、inetd(試作は Solaris10

```
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.0.0.94.0.1.53 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.0.35.50.211.8.82 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.0.35.71.22.224.192 = INTEGER: 0
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.0.36.29.33.178.118 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.0.36.232.15.212.31 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.92.255.53.2.88.244 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.96.235.105.217.81.212 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.96.235.105.217.82.213 = INTEGER: 3
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.112.113.188.74.77.163 = INTEGER: 24
SNMPv2-SMI::mib-2.17.7.1.2.2.1.2.1.164.103.6.179.123.136 = INTEGER: 24
```

図 10 SNMPwalk の出力例

で行ったので実際には smf) に登録する。今回の試作ではシェルと awk *1 で実装した。

lansockrep は設定ファイルを元に snmpwalk で MIB .1.3.6.1.2.1.17.7.1.2.2.1.2 を取得し、MAC アドレスを検索する。発見した場合はハブのホスト名(または IP アドレス)と、受信したデータを合わせて 1 行としてログファイルに追加する。このログファイルを事後に処理すれば対応表が作成できる。

FDB の取得であるが、現状では Q-Bridge MIB⁴⁾ の FDB だけを逐次的に問い合わせる簡易実装となっている。

一連の処理の終了後、lansockrep は終了し、クライアントとの TCP 接続が切断される。

HP procurve switch 2810-24G に対して snmpwalk で問い合わせを行った場合の出力例を図 10 に示す。= の直前の 10 進表記 6 個が MAC アドレス。行末の整数値がポート番号である。実際のポートに対応している番号は 1~24 である。24 番がアップリンク側で、機器が接続されているのは 3 番である。0 番に対するデータは、ハブ自身の MAC アドレスである。端末から送られてくる mac アドレスはコロン区切りの 16 進表記なので比較に先立って変換が必要である。

サーバーの /var/log/lansockrep には図 11 に例示する書式で調査結果が書き込まれる。これを処理して整理すると図 12 のような情報コンセント識別子-ハブポート番号対応表が得られる。

*1 sprintf による 16 進変換機能が必要であるので、nawk または gawk が必要

```
Fri Feb 10 16:54:39 JST 2012 found: A1 4106-01 ca:d8:4e:03:00:08 10.45.17.109 15
Fri Feb 10 17:03:50 JST 2012 found: A1 4106-02 ca:d8:4e:03:00:09 10.45.17.109 16
Fri Feb 10 17:04:30 JST 2012 found: A1 4106-03 ca:d8:4e:03:00:0a 10.45.17.109 17
Fri Feb 10 17:04:44 JST 2012 found: A1 4106-04 ca:d8:4e:03:00:0b 10.45.17.109 18
Fri Feb 10 17:05:28 JST 2012 found: A1 4106-05 ca:d8:4e:03:00:0c 10.45.17.109 19
Fri Feb 10 17:07:23 JST 2012 found: A1 4106-15 ca:d8:4e:03:00:1e 10.45.17.108 15
Fri Feb 10 17:07:37 JST 2012 found: A1 4106-14 ca:d8:4e:03:00:1f 10.45.17.108 14
Fri Feb 10 17:09:52 JST 2012 found: A1 4106-13 ca:d8:4e:03:00:20 10.45.17.108 13
Fri Feb 10 17:10:13 JST 2012 found: A1 4106-12 ca:d8:4e:03:00:21 10.45.17.108 12
Fri Feb 10 17:10:52 JST 2012 found: A1 4106-11 ca:d8:4e:03:00:22 10.45.17.108 11
```

図 11 得られるログファイルの例

4106-01 10.45.17.109 15	4106-11 10.45.17.108 11
4106-02 10.45.17.109 16	4106-12 10.45.17.108 12
4106-03 10.45.17.109 17	4106-13 10.45.17.108 13
4106-04 10.45.17.109 18	4106-14 10.45.17.108 14
4106-05 10.45.17.109 19	4106-15 10.45.17.108 15

(右段に続く)

図 12 ログファイルから抽出した対応表

3.5 動作実験

表 1 の実験環境で動作実験を行った。

1つの情報コンセントに対して enter キーを押してから調査が終了するまでの所要時間は 6 秒程度であったが、時折 10 数秒を要することがあった。内訳は以下の通りであった。

3~4 秒... ifconfig up

2~10 数秒... dhcp による IP アドレス割り当て

1 秒未満... サーバによる FDB 探索その他

snmpwalk の所要時間は L2SW の個数 × 0.2 秒程度であった。100 席 (情報コンセント 100 口) 程度の教室を収容するエッジスイッチは 3~4 基であるので、実際の環境で SNMP 問い合わせの所要時間が問題になることは無いと考えられる。

試作した調査クライアント端末はテンキーの大きさが支配的であるので、その代わりに片手に握って操作可能なゲームコントローラなどを採用することで作業が効率化できると考えられる。

表 1 実験機材

L2SW	HP procurve switch 2810-24G HP procurve switch 2610-24
調査サーバ	Sun Javaworkstation2100(solaris10/Sun OS5.10)
調査クライアント端末	Armadillo-220(ユーザーランド atmark-dist 20111220, linux カーネル 2.6.12.3-a9-18)

4. まとめ

本研究では、情報コンセントのポートに対応するハブとポート番号を調査し、作表する作業を支援するシステムを開発した。

電池動作の小型調査端末を順次情報コンセントに接続して、ハブの FDB から調査したポート対応関係をサーバに蓄積するシステムである。サーバのログを処理することで対応表が作表できる。

参考文献

- 1) 河野優他: ループを考慮した Layer2 ネットワーク構成情報の推測アルゴリズムについて, 分散システム/インターネット運用技術シンポジウム 2006 年度論文集, Vol.2006, No.13, pp.7-12(2006).
- 2) David T Stott: Layer-2 Path Discovery Using Spanning Tree MIBs, Knowledge Creation Diffusion Utilization, Volume 004, Citeseer(2002).
- 3) E. Decker, P. Langille, A. Rijhsinghani and K. McCloghrie: Definitions of Managed Objects for Bridges, RFC1493(1993).
- 4) E. Bell, A. Smith, P. Langille, A. Rijhsinghani and K. McCloghrie: Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions, RFC2674(1999).
- 5) Brad Hards: The Linux USB Input Subsystem, Part I, <http://www.linuxjournal.com/article/6396>, LINUXJournal(2003).
- 6) Brad Hards: The Linux USB Input Subsystem, Part II, <http://www.linuxjournal.com/article/6429>, LINUXJournal(2003).
- 7) ると: Linux Input Subsystem の使い方, [http://takuo.sakura.ne.jp/\\$sim\\$takuo/input_subsystem/input_subsystem.html](http://takuo.sakura.ne.jp/simtakuo/input_subsystem/input_subsystem.html)(2012).
- 8) Denys Vlasenko: BusyBox: The Swiss Army Knife of Embedded Linux, <http://busybox.net/about.html>