



Village Telco
Documentation
version 0.2

Table of contents

Village Telco.....	4
Vision	4
The service.....	4
The technology	4
Mesh networking	4
Voice over Internet Protocol	5
The elements.....	5
The client	5
MeshPotato	5
Server	6
A2Billing.....	6
Afrimesh	6
A3Glue	7
Scenarios	7
Stand-alone installation	7
Telephony installation.....	7
Internet Installations	7
Combined Internet and Telephony installation	7
Installation.....	8
Installing the MeshPotato	8
Physical installation	8
Equipment needed for installation.....	8
Configuring the MeshPotato with a phone	8
Connecting with a network cable.....	9
Fresnel zone – the tunnel of the signal	9
The lights	9
Power the MeshPotato	9
Use both telephony and network to the MeshPotato	12
Reflashing the MeshPotato (the hard way).....	12
Installing the VillageTelco Server	12
Hardware.....	13
Software	13
Village Telco Server Installation	13

What is it?.....	13
Components	13
Installation.....	14
Pre-requirements	14
Download DEB package	14
Verify the MD5 signature	14
Install the villagetelco-server	14
First steps	16
SVN checkout.....	16
Trunk and Tagged versions.....	16
The Village Telco Wizard	16
User Guides	20
Afrimesh	20
A2Billing.....	20
Make more vouchers.....	20
Troubleshooting	20
Debugging MeshPotato installations	20
No power	20
The dial-tone is choppy	20
Choppy sound.....	20
The MeshPotato is flashed with AP51.....	20
Looking at the routing tables	22
Rescue TFTP of the MeshPotato	23
Using the network scanner.....	23
Debugging the Village Telco Wizard.....	23
Appendix.....	24
Appendix A – Specifications on the MeshPotato	24
Appendix B – Setting up automatic ssh into all mesh potatoes.....	25
Acronyms.....	26
References.....	26

Village Telco

Vision

“The Village Telco aims to provide affordable voice and data services for the billions of people who are currently unconnected to either a phone or the Internet. By lowering the cost of entry for users, it also seeks to make calling and access to the Internet cheaper for existing users. But it will not only offer affordable access but put the power to provide it in the hands of local communities, creating a new generation of communications entrepreneurs in developing countries”.

The service

The Village Telco will provide a “plug-and-play” low cost voice and data service delivered using an inexpensive, fixed mesh Wi-Fi delivery system called the Mesh Potato.

For the consumer, the basic service will give the customer pre-paid local, national and international calling and Internet. Other services like SMS and voice message boxes can be added. It will both provide a service where none exists and be a complement to mobile services. Limited mobility exists as an option but will not necessarily be deployed in order to avoid head-on confrontations with powerful mobile operators.

For entrepreneurs in developing countries, it offers the opportunity to start a business that will provide them with a livelihood and contribute to the development of their communities. Again, the network required to deliver it will be “plug-and-play” and be robust enough to withstand a variety of challenging circumstances.

For the developers of the concept, it is the opportunity to create a global network of low-cost voice and data providers.

Whilst there are a range of other Wi-Fi mesh products on the market, the Mesh Potato has two features that make it stand out: 1) It combines voice and data seamlessly; and 2) It will be a “plug and play” product that will allow consumers to use it more or less “out-of-the-box”.

The technology

Mesh networking

Mesh networking is a type of networking wherein each node in the network may act as an independent router, regardless of whether it is connected to another network or not. It allows for continuous connections and reconfiguration around broken or blocked paths by “hopping” from node to node until the destination is reached. A mesh network whose nodes are all connected to each other is a fully connected network. Mesh networks differ from other networks in that



the component parts can all connect to each other via multiple hops, and they generally are not mobile. Mesh networks can be seen as one type of ad hoc network.

Mesh networks are self-healing: the network can still operate when one node breaks down or a connection goes bad. As a result, the network may typically be very reliable, as there is often more than one path between a source and a destination in the network.

Voice over Internet Protocol

Voice over Internet Protocol (Voice over IP, VoIP) is a general term for a family of methodologies, communication protocols, and transmission technologies for delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Other terms frequently encountered and often used synonymously with VoIP are IP telephony, Internet telephony, voice over broadband (VoBB), broadband telephony, and broadband phone.

Internet telephony refers to communications services — voice, fax, SMS, and/or voice-messaging applications — that are transported via the Internet, rather than the public switched telephone network (PSTN). The steps involved in originating an VoIP telephone call are signaling and media channel setup, digitization of the analog voice signal, optionally compression, packetization, and transmission as Internet Protocol (IP) packets over a packet-switched network. On the receiving side similar steps reproduce the original voice stream.

VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. Codec use is varied between different implementations of VoIP (and often a range of codecs are used); some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs.

The elements

The client

MeshPotato

The current Mesh Potato is an 802.11bg mesh router with a single FXS port and one 10/100Mbit Ethernet port. Adjacent Mesh Potatoes automatically form a peer-to-peer network, relaying telephone calls without landlines or cell-phone towers. The Mesh Potato hardware and software is open. The power, Ethernet and FXS ports are robust in order to deal with developing-world conditions like static electricity, lightning, bad power and accidental abuse. It comes in a weatherproof box for outdoor mounting and costs about the same as any other Wi-Fi router.



An analogue phone connects to the Mesh Potato via the FXS port. FXS (Foreign eXchange Station) is a telephone interface that supplies power, dial tone and generates ringing voltage. When you make a

phone call, your Mesh Potato talks to the Potato down the street, which talks to the next Potato, and eventually to the destination. The mesh network can be augmented via backbone links and connected to the rest of the world using VoIP trunks.

The development team plan to do extensive network testing with the existing Mesh Potato and develop a similar next generation device with an 802.11n chip. This will increase the performance of the Mesh Potato, making it somewhere between 2-4 times faster and at least double the range of the device.

Server

Only one computer is required to provide the services for the Village Telco. This computer, called the VT Server, could be anything from a standard laptop to a high-end server for better redundancy. The services on this computer are made up of several elements including A2Billing, Afrimesh and A3Glue. This can be run on a standard laptop computer or you can go for a more high-end server for better redundancy. The server part consists of several software elements.

A2Billing

A2Billing combined with Asterisk is a full featured telecom platform and softswitch providing converged services, with self contained billing (pre or post-paid), reporting and statistics for IP and TDM based voice networks and can be configured to supply a wide range of services, rate calls, prepare and send out invoices, as well as accept payments via a number of payment service providers.

The A2Billing solution comprises of the following components:

- Server: The computer to run the system.
- Line Interface Cards: The Digium hardware (optional) to connect to the TDM network (PRI, BRI, Analogue)
- Linux: The base operating system
- Asterisk: The telephony engine
- Apache: The web server
- MySQL/Postgresql: The back end database
- A2Billing: The Billing engine handling Authentication, Authorisation and Accounting.

Afrimesh

Building on the innovative research efforts and tireless support of the Wireless Africa group at the Meraka Institute of the CSIR, the sometimes terrifying growth of the Scarborough Wireless Users Group and tools developed by the Open Mesh team Afrimesh provides a simple management dashboard helping network operators create and sustain resilient communications networks with a minimum of fuss.

Featuring

- Powerful mesh network routing with B.A.T.M.A.N.
- Dynamic GIS visualization of your mesh on OpenStreetMap maps.
- Plan your network using the terrain elevation map.
- Visualize live health&traffic information for any mesh node.

- Keep inventory of your network devices.
- Monitor the health&usage of your Internet gateway.
- Monitor live network accounting information from pmacct.
- View network status & log messages in realtime.
- Customer management interface featuring FreeRADIUS support.
- Customer authentication featuring coova-chilli support.
- Online help and live chat support.
- Intelligent configuration assistance.
- Build and manage mesh networks even without an Internet connection.
- Now supports the VillageTelco Mesh Potato.

A3Glue

This is the glue, making Afrimesh, A2billing and the Mesh network work together. It gathers data from the network, provisioning the MeshPotatoes and make sure everything runs smoothly.

Scenarios

Stand-alone installation

Mesh potatoes can be deployed from two units and you can make calls between them. You can then add more units to the network one at a time. You will be able to make calls and can have a closed network between the units.

Telephony installation

Deploy the MeshPotato for telephony and add a SIP Provider or PSTN connection to the A2Billing server. This will allow for free calls between the units and cheaper international calls.

Internet Installations

Using the MeshPotato to provide Internet to the Village. The Internet traffic goes out through a common gateway. The community needs to pay for an Internet connection through an Internet Service Provider and share the costs between its members.

Combined Internet and Telephony installation

Deploy MeshPotatoes and use it both for Internet and telephony for the Village. You connect an Internet gateway for shared broadband for the users and telephony connection to call international.

Installation

Installing the MeshPotato

The MeshPotato using WLAN to connect to each other on the 2,4GHz spectrum. They connect to each other with mesh networking. This 2,4GHz spectrum is free in most countries all over the world, license free.

Every MeshPotato should have free line of sight (LOS) to at least one more MeshPotato. The higher the installation of the MeshPotato, normally the better the signal. The more MeshPotatoes that can hear each other, the network gets stronger and more redundant.

Physical installation

The MeshPotato comes with a bracket for either wall or mast mount. The wall mount takes two screws and then you slide the MeshPotato in place. The Mast mount takes two strips to tighten it to the mast.

Equipment needed for installation

You need the following tools to install the MeshPotato:

1. Cable
 - Cat 5 cable made for outdoor installation (UV proof)
 - RJ14/RJ45 crimping tool
 - Wire cutter

Tip: A 12Volt lead-acid sealed battery with the right plug gives you the option to test each mounting point before fastening the device.

Step by step

1. Before: Crimp on end on the cable, ready for deployment.
2. Check the MeshPotato with the power supply inside the house
3. Find the best spot for the MeshPotato on a high-point.
4. Run the cable down and put a cable on the other end
5. Connect local phone (and/or computer)
6. Configure the device and reboot it (power cycle it)
7. Make a test call.

Configuring the MeshPotato with a phone

The MeshPotato has an integrated interactive voice menu. Plug in power and an analogue phone, dial C O N F (2663). You will hear the following menu:

Press the full IP address of the device, using * for the dots (.). Example 10*130*1*50#. You will then hear the IP address being repeated back to you. Power cycle the device after you have heard that you have set the right address.

Connecting with a network cable

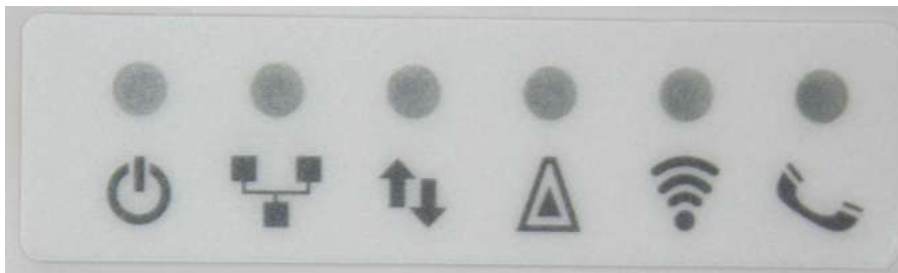
The MeshPotato comes with default IP 192.168.1.20 (in a /24 – C network) on the Ethernet port. It also has an IP of 172.31.255.254/30, where you need to set your PC to 172.31.255.253 (/30 or network mask 255.255.255.252)

Fresnel zone – the tunnel of the signal

The MeshPotato can run for 1-2 kilometers if installed in a high place with clear line of sight. The signal from one MeshPotato to another travels in a “tunnel” between each other. XXX PICTURE

The lights

The MeshPotato has five lights indication the status of the MeshPotato.



First from the left: Power. (should be steady red colour)

Second from left: Ethernet connection (steady colour when link to a computer)

Third from left: Traffic (steady on, and flashing when traffic from Ethernet)

Fourth from left: (Not in use at the moment)

Fifth: Wireless connection. (Blinking steadily when enabled)

Last on the right: Telephony (Not in use at the moment)

Power the MeshPotato

The MeshPotato can handle from 9 to 40 volt and it can be powered in three ways, through a regular power supply, through the telephone cable or through the network cable.

WARNING: Only use one power connection in any one time.

Use a regular power supply

It comes with a 12V power supply which handles 100 to 240 volt. The plug is EU standard.


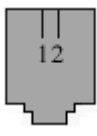



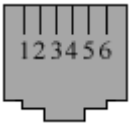
As an alternative, a separate 9V to 42 Vdc source may be connected via a DC power plug with a 5.5 mm diameter shaft with 2.1 mm diameter centre hole (XXX to be confirmed) and 9 mm length. The centre pin is positive and the outer is the common ground of the MeshPotato.

WARNING: Note that excessive supply voltage can damage the MeshPotato in some circumstances.

Use the telephone cable to run power





The MeshPotato has a 6 pin. The first 2 pins is the plus, the two in the middle is the phone connection and the two last one is the minus. Use Orange for +(pluss), Blue for phone, Orange for – (minus).



Have as large diameter (thick leaders) on the cable if you are running a long cable (10meter+). The voltage drop will be lower (remember you have both voltage drops up to the MeshPotato, and back again on the telephony/FXS port).

Type	Picture	Cable lay-out	Cabling	Comments
RJ11 ONLY for TELEPHONY!			Connect one pair on the two middle pins.	Can only be used for telephony, NOT power!
RJ14 (NOT RECOMMENDED!)			First pair of wire on pin 1 and 4 (for power). Second pair on wire 2 and 4 (for phone)	This can not be longer than maximum 8 metres because of voltage loss in the cable
RJ25 RECOMMENDED for telephony and power			First pair (orange) on pin 1 and 2. Second pair on pin 3 and 4 (blue) Third pair on 5 and 6 (brown)	

Function	RJ-11 pin	RJ-14 pin	RJ-25
+12 Vdc input			1
+12 Vdc input		1	2
Phone ('ring' / -)	1	2	3
Phone ('tip' / +)	2	3	4
0 Vdc input (MP01 Gnd)		4	5
0 Vdc input (MP01 Gnd)			6

Preferred wiring

Function	RJ- pin	Colour schema (TIA/EIA 568A Wiring)
+12 Vdc input	1	 White and Orange
+12 Vdc input	2	 Orange
Telephony	3	 White and Blue
+12 Vdc input	4	 Blue

0 Vdc input (MP01 Gnd)	5		White and Brown
0 Vdc input (MP01 Gnd)	6		Brown











Use the network cable to run power

The network cable is using 4 cables (2 pairs) for the Ethernet connection. The two other pairs are used for running the power.



Cabling: Ethernet plug uses pin 1, 2, 3 and 6 (yes, sixth pin) for the networking and pin 4, 5, 7 and 8 is used for power.

Function	RJ-45 pin	Colour schema (TIA/EIA 568A Wiring)
Networking RX+	1	 White and Orange
Networking RX-	2	 Orange

Networking TX+	3		White and Green
+12 Vdc input	4		Blue
+12 Vdc input	5		White and Blue
Networking TX-	6		Green
0 Vdc input (MP01 Gnd)	7		White and Brown
0 Vdc input (MP01 Gnd)	8		Brown

Use both telephony and network to the MeshPotato

The best solution is to run two cables to the MeshPotato. One for telephony and power, one for Ethernet.

If you only can run one single cable and it is over 8metres, you need to have a power supply giving out at least 20Vdc. This is because of the loss of voltage in a small cable.

Refashing the MeshPotato (the hard way)

1. Set your PC to IP 192.168.1.185 (netmask 255.255.255.0 as a /24 network)
2. Set up a TFTP server with files from:
<http://downloads.open-mesh.org/mesh-potato/>
3. Plug a network cable from your PC to the MeshPotato
4. Be ready to telnet to 192.168.1.20 on port 9000
5. Ping 192.168.1.20
6. When it responds, start the telnet session to port 9000 (you only have 5 seconds approximately 4 seconds after powering on the MeshPotato)
7. You will get the following connection <RedBoot>
8. Run the following procedures
(NB, don't push any keys in the session while working, this will time-out the session)
 - a. fis init
(10 seconds)
 - b. load -r -b % {FREEMEMLO} openwrt-atheros-vmlinux-rv248.lzma
(5 seconds)
 - c. fis create -e 0x80041000 -r 0x80041000 vmlinux.bin.l7
(approx 100 seconds)
 - d. load -r -b % {FREEMEMLO} openwrt-atheros-root-rv248.squashfs
(5 seconds)
 - e. fis create rootfs
(8 minutes(!) be patient!)
 - f. Power cycle the unit

Installing the VillageTelco Server

The installation of the VillageTelco Server is easy.

Hardware

You need a standard PC with at least 1GHz CPU, 256Mbyte memory and a 10 gigabyte hard drive. The hardware must be supported by Ubuntu 8.04 or 10.04 distributions.

For using the server as an Internet or telephony gateway, it should have two Ethernet connections (minimum 100Mbit).

Tested hardware:

Acer One – netbook. It is a 1.6GHz CPU and 1 gig memory with a 160Gbyte hard drive. The positive about a netbook is the integrated battery, enabling it to survive shorter power outages and will controllable shut down the server if it is low on battery. It was used an Ethernet USB dongle, but there was problem with the latest kernel, but a roll-back to a previous one fixed the issue.

Software

Operating system

The server is based on Ubuntu version 8.04 LTE or 10.04 LTE. Before starting the installation of the Village Telco server, remember to run a full update on the packages (Command: `sudo apt-get upgrade`)

Installation of Ubuntu Server 10.04 LTE

Obtain the ISO image from the net and transfer it to a CD or USB stick.

WARNING: When installing the Ubuntu distribution, it will overwrite all data on the computer. Please use a dedicated computer for this purpose.

Village Telco server

Village Telco Server Installation

What is it?

It is a .deb package that installs in both 8.04 and 10.04 Ubuntu distributions. It contains all the necessary software to run a billing server (a2billing) for Village Telco. The package also includes an installation wizard (wizard) and a monitoring/provisioning service (a3glue)

Components

The Village Telco Server package is composed of three main components:

- **a2billing village telco** : Based on a2billing version 1.7.0, a2biling incorporates our Village Telco simplified GUI, a SOAP webservice and a few extra patches to work in Ubuntu 10.04. A2billing is written in PHP+smarty and interfaces with Asterisk using AGI and AMI.
- **a2billing installation wizard** : The wizard interfaces with the SOAP webservice and performs a full installation of a2billing for Village Telco in 5 steps. The wizard is written in PHP+Cake MVC and interfaces with a2billing with SOAP.

- **a3glue** : The a3glue is a webservice that follows the REST architecture and delivers JSON provisioning and monitoring information to other components of the Village Telco as the Afrimesh monitoring panel. A3glue is written in PHP and interfaces with Asterisk using Asterisk AJAM (MXML).

Installation

This document describes how to install the development package of villagetelco-server. Latest version is always available at: <http://dev.villagetelco.org/debs/>

LATEST VERSION 0.31-18 - 18th September 2010

Pre-requirements

1. Install the pre-require packages as follows:

```
apt-get install libapache2-mod-php5 php5 php5-common php5-cli \  
php5-mysql mysql-server apache2 php5-gd php5-curl php5-mcrypt php-soap php-pear asterisk
```

This is around 150 MB in a ubuntu 8.04.

IMPORTANT When you are prompted for the root password of mysql (New password for the MySQL "root" user:) Make sure that you do not forget the password as you will need it later! If you want just to test, you can press ENTER as root password for mysql. i.e.: no password.

Asking for the MySQL "root" password up to three times is normal during this step!

Download DEB package

Change what is the last version :-) for 0.31-18

```
cd /var/tmp; wget http://dev.villagetelco.org/debs/villagetelco-  
server_0.31-18_all.deb  
cd /var/tmp; wget http://dev.villagetelco.org/debs/villagetelco-  
server_0.31-18_all.deb.md5
```

Verify the MD5 signature

```
md5sum /var/tmp/villagetelco-server_0.31-18_all.deb
```

Install the villagetelco-server

```
cd /var/tmp/; dpkg -i villagetelco-server_0.31-18_all.deb
```

IMPORTANT Until we create alternative asterisk-config packages, we are making backup copies of the files we overwrite in /etc/asterisk and /usr/share/asterisk/agi-bin We expect a apache2 fresh installation. We are disabling the *default* website and enabling a new website *villagetelco* under /etc/apache2/sites-enabled

If it is the first time you install the package you can answer "N" to the question of dropping the databases. If you have installed a previous version answer "y" to the question Do you really want to drop the 'mya2billing a2b_wizard a3glue' database?

```
root@foo-desktop:/var/tmp# dpkg -i villagetelco-server_0.31-18_all.deb
(Reading database ... 144022 files and directories currently installed.)
Preparing to replace villagetelco-server 0.31-18 (using villagetelco-
server_0.31-18_all.deb) ...
Unpacking replacement villagetelco-server ...
Setting up villagetelco-server (0.31-18) ...
Villagetelco post-installation
[general conf] Placing a2billing.conf in /etc
[asterisk additional conf] adding sip, iax additional conf
[asterisk main backup] Asterisk backup of orig sip, iax, http, manager,
extensions conf
[asterisk main conf] A2B conf sip, iax, http, manager, extensions conf
[asterisk sounds] extra english sounds for A2B
[asterisk AGI] agi-bin scripts for A2B
[GUI and A2B webservice]
[A2B, wizard and a3glue db init] Creating databases, users and populating
schemas
mysql root password:

Dropping mysql databases
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'mya2billing' database [y/N] y
Database "mya2billing" dropped
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'a3glue' database [y/N] y
Database "a3glue" dropped
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'a2b_wizard' database [y/N] y
Database "a2b_wizard" dropped
Creating users and new schemas
[apache2] Creating domain and rewrite rules for services
[misc] changing memory_limit in PHP
[apache2] Enabling villagetelco services
Module rewrite already enabled
Site default already disabled
Site villagetelco already enabled
[monitoring]
[logging]
[cron jobs] a2glue cache server, currency service
[services] restarting services
* Reloading web server config apache2    [ OK ]
Stopping Asterisk PBX: asterisk.
Starting Asterisk PBX: asterisk.
cron start/running, process 12644
=====
Village Telco Server Installed!!!
Installation is available under /usr/local/villagetelco-server
Backup files of original configurations are available
/etc/asterisk/*.1284810055 /usr/share/asterisk/agi-bin.1284810055
```

Get started!!!
Create a first A2B installation by running the wizard
<http://localhost/wizard>

Log into A2B to see your users and vouchers
<http://localhost/a2billing>

Afrimesh developers have a monitoring client available at
http://localhost/a3glue/a3g_cli.php

First steps

1. Run the wizard

Go to: <http://localhost/wizard/>

You will log into the wizard with user: root pass: changepassword First thing to do is to change the root password!

2. Go to a2billing

Go to: <http://localhost/a2billing>

3. Run asterisk client and reload asterisk to see all the SIP peers

asterisk -r

```
villagetelco804*CLI> sip show peers  
villagetelco804*CLI> sip reload  
villagetelco804*CLI> sip show peers
```

4. Check /var/tmp/ for logs from a3glue to see the cache server working

SVN checkout

Trunk and Tagged versions

trunk: `svn co http://dev.villagetelco.org/svn/villagetelco/server/trunk`

tags-0.32-11: `svn co`

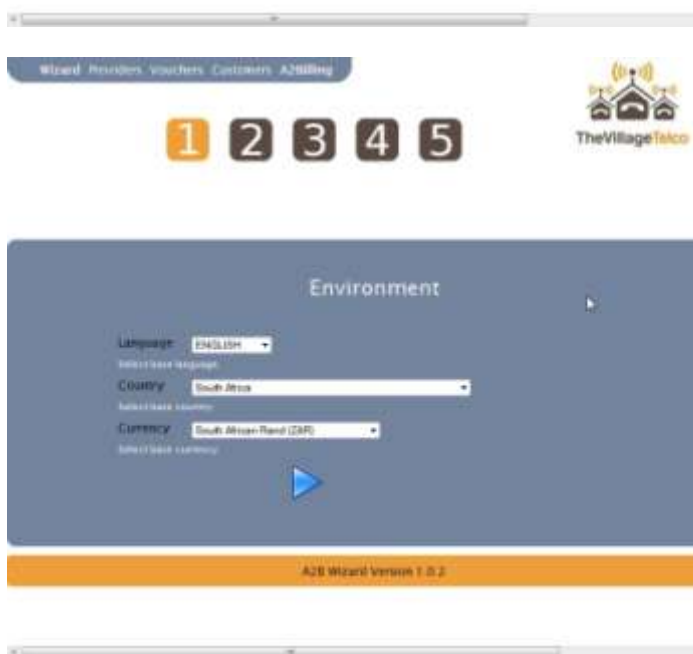
`http://dev.villagetelco.org/svn/villagetelco/server/tags/0-32.11`

The Village Telco Wizard

You have now installed the Village Telco server.

Go to <http://<address of the server>/wizard>

Log in as "root" with password "changepassword"





- 1
- 2
- 3
- 4
- 5

Credit and Rates

Account balance SAR
State the initial balance of each customer account

Local rate SAR/min
State the rate to be applied to local phone calls

Navigation: < >

AZB Wizard Version 1.0.2



- 1
- 2
- 3
- 4
- 5

Vouchers

Credit: Item:

Credit: Units:

Add vouchers

Navigation: < >

AZB Wizard Version 1.0.2

Wizard: Providers, Vouchers, Customers, A/Billing

1 2 3 4 5

TheVillageTelco

Voucher Design

Title:

Write the text that you want to appear on the vouchers

Voucher Logo

-
-
-
-
-
-
-
-
-
-

← →

A2B Wizard version 1.0.2

Wizard: Providers, Vouchers, Customers, A/Billing

TheVillageTelco

Wizard Overview

General

Language	en
Country	ZAF
Currency	ZAR
Number of customers	100
total customer balance	20 ZAR
Local prefix	111
Rate for local calls	1 ZAR

Vouchers

Credit	Units	Total
10 ZAR	20	200 ZAR
100 ZAR	50	5000 ZAR
Total amount		5200 ZAR

Voucher Design

Title:

Logo:

A2B Wizard version 1.0.2

User Guides

The normal day procedures to sustain a smaller or larger wireless and VoIP network.

Afrimesh

A2Billing

Make more vouchers

Troubleshooting

There are a lot of tools to help troubleshoot different situations happening.

Debugging MeshPotato installations

Debug diagram XXX ?

No power

Please check that it is power in the socket by putting another device.

Change the power supply to another one.

The dial-tone is choppy

You just installed the mesh potato but even before doing a call, the dial-tone is choppy and not a consistent tone. The MeshPotato networking can run down to 5 volt, but the analog port needs at least 9 volt. Check the power is good and if you are running a very long cable to the MeshPotato, either get a better cable (bigger diameter) or increase the voltage feeding it (you are probably feeding it with 12 volt.) Increase the power supply to 20+ volt. Also check that you are using four cables to run the power to the MeshPotato, one pair for plus and one pair for minus.

Choppy sound

This is most likely caused by a poor wireless link. Check that you have line of sight to other MeshPotatoes. It is also possible to install another MeshPotato on a nearby building or tower to relay the signal.

The MeshPotato is flashed with AP51

The AP51 flash tool re-arranges the disk lay-out. If the MeshPotato has been updated with the AP51, you CAN NOT upgrade it remotely through wireless. Then you need to do a MeshPotato Disaster Recovery BEFORE deploying it in the field to be able to remotely upgrade it in the future.

CORRECT FORMAT:

```
BusyBox v1.14.4 (2009-12-11 15:17:30 CET) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
|_____|.-----|_____|.-----|_____|.-----|_____|.-----|_____| | | | |
|  -  ||  _  |  -  ||  |  |  ||  _  |  |  |  |
|_____| ||  _  |  _  |  _  |  _  |  _  |  _  |  _  |
      |__| W I R E L E S S   F R E E D O M
KAMIKAZE (bleeding edge, r203) -----
```

```

* 10 oz Vodka          Shake well with ice and strain
* 10 oz Triple sec    mixture into 10 shot glasses.
* 10 oz lime juice    Salute!

```

```

-----
root@OpenWrt:~# cat /proc/mtd
dev:   size  erasesize  name
mtd0: 00030000 00010000  "RedBoot"
mtd1: 000b0000 00010000  "vmlinux.bin.l7"
mtd2: 006f0000 00010000  "rootfs"
mtd3: 00410000 00010000  "rootfs_data"
mtd4: 0000f000 00010000  "FIS directory"
mtd5: 00001000 00010000  "RedBoot config"
mtd6: 00020000 00010000  "boardconfig"
root@OpenWrt:~#

```

The ap51-utility swaps the position of kernel and rootfs. The kernel needs to be in the second logical block and the rootfs in the third. After flashing with ap51-flash this order is reversed and any flashing attempt from OpenWRT will fail.

Before flashing with ap51-flash:

```

root@Mesh-Potato:/# cat /proc/mtd
dev:  size  erasesize  name
mtd0: 00030000 00010000  "RedBoot"
mtd1: 000b0000 00010000  "vmlinux.bin.l7"
mtd2: 006f0000 00010000  "rootfs"
mtd3: 00410000 00010000  "rootfs_data"
mtd4: 0000f000 00010000  "FIS directory"
mtd5: 00001000 00010000  "RedBoot config"
mtd6: 00020000 00010000  "boardconfig"

```

After flashing with ap51-flash:

```

RedBoot> fis list
Name          FLASH addr Mem addr  Length  Entry point
RedBoot       0xA8000000 0xA8000000 0x00030000 0x00000000
rootfs        0xA8030000 0xA8030000 0x006F0000 0x00000000
vmlinux.bin.l7 0xA8720000 0x80041000 0x000B0000 0x80041000
FIS directory 0xA87D0000 0xA87D0000 0x0000F000 0x00000000
RedBoot config 0xA87DF000 0xA87DF000 0x00001000 0x00000000

```

The fix is to flash the system via Redboot with the correct layout and follow the instructions here:

http://wiki.villagetelco.org/index.php?title=Mesh_Potato_HOWTOs#Reflash_the_firmware_from_RedBoot.

Unfortunately setting up atftpd is not easy - we ran into several issues with Ubuntu. Here is a primer of the issues we encountered:

* Install putty and atftpd (and maybe atftp to check whether atftpd is working by connecting locally)

* Stop the network-manager and configure the ethernet port on the command line:

```
ifconfig eth0 192.168.1.185 (use this IP as the production MP expect the tftp server here)
```

* Remove opensbsd-inetd after installing atftpd using aptitude, otherwise atftpd can not run standalone

* Place the openwrt-atheros* files into /tftpboot, be generous with permissions (chmod -R 555 /tftpboot/) and start atftpd this way:

```
atftpd --daemon /tftpboot
```

* ping 192.168.1.20 in one console (keep it running)

* start putty and configure it to connect to 192.168.1.20 using the telnet protocol and port 9000

* power cycle the MP and connect to the MP as soon as you get the first ping replies

Follow the procedure documented in the wiki

Looking at the routing tables

You can check the routing table.

Default routing table (not the batman routing table)

```
root@Mesh-Potato:/# ip r
172.31.255.252/30 dev eth0 proto kernel scope link src 172.31.255.254
10.130.1.0/24 dev ath0 proto kernel scope link src 10.130.1.39
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.20
default via 192.168.1.20 dev eth0 scope link
```

For the full batman routing table, type:

```
root@Mesh-Potato:/# ip r ls table 66
10.130.1.17 via 10.130.1.32 dev ath0 proto static src 10.130.1.39
10.130.1.32 dev ath0 proto static scope link src 10.130.1.39
10.130.1.16 dev ath0 proto static scope link src 10.130.1.39
10.130.1.33 dev ath0 proto static scope link src 10.130.1.39
10.130.1.34 dev ath0 proto static scope link src 10.130.1.39
10.130.1.20 dev ath0 proto static scope link src 10.130.1.39
10.130.1.24 via 10.130.1.34 dev ath0 proto static src 10.130.1.39
10.130.1.29 dev ath0 proto static scope link src 10.130.1.39
10.130.1.12 via 10.130.1.16 dev ath0 proto static src 10.130.1.39
10.130.1.28 dev ath0 proto static scope link src 10.130.1.39
```

10.130.1.31 dev ath0 proto static scope link src 10.130.1.39
10.130.1.30 dev ath0 proto static scope link src 10.130.1.39

Rescue TFTP of the MeshPotato

If just the boot loader is responding, you need to do a rescue with TFTP transfer.

When booting the MeshPotato, you will see that you get ping response from 192.168.1.20 approx 10 seconds after booting and only 5-8 replies before it boots the full kernel. This is the boot loader responding.

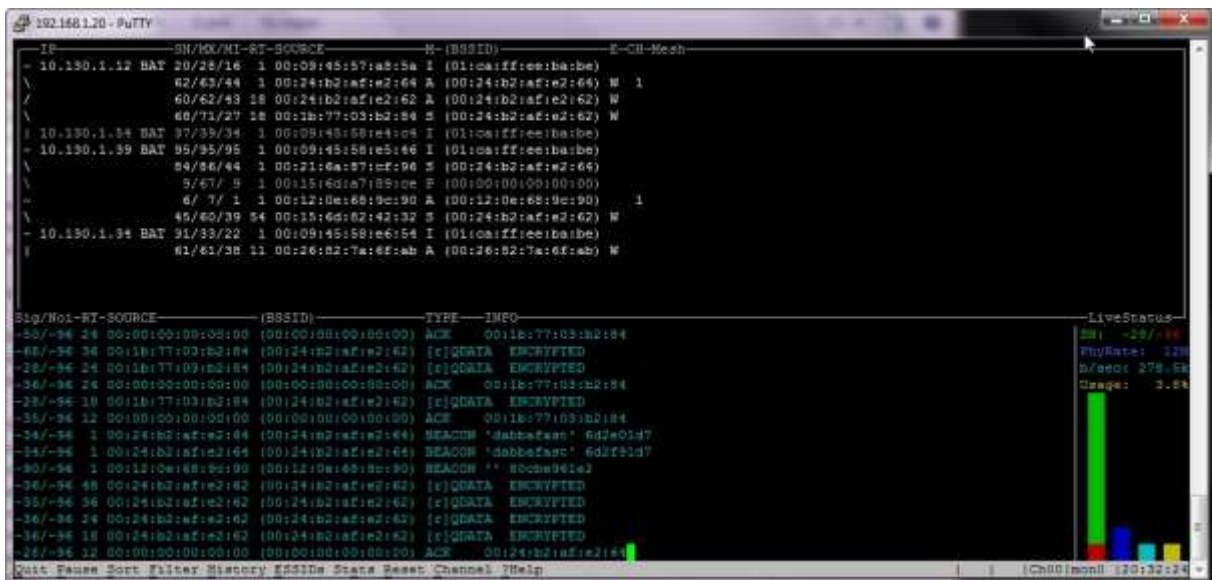
Using the network scanner

Horst is a built in network scanner.

WARNING: If it is a lot of mesh devices nearby, this may hang your MeshPotato! Be careful running it remotely.

Run:

root@Mesh-Potato:/# horst.sh



The upper part shows the mesh nodes it can see.

The lower left shows live packets running on the wireless network.

The lower right shows the following:

- SN: Signal to Noise ratio (Signal =Green, Noise = red colour)
- PhyRate: Physical transferring rate
- b/sec: Bytes per second running on the wireless network
- Usage: How much of the wireless channel (available bandwidth)that is in use.

Debugging the Village Telco Wizard

Go to <http://<server ip address>/wizard/log>

Here will the log of what the wizard did. Please check that you have a valid Internet connection while doing the wizard since it relies on exchange rates fetched from the Internet.

Appendix

Appendix A – Specifications on the MeshPotato

Feature	Benefit
Integrated Wifi and ATA in a single box	Low power consumption, ease of setup, fewer cables, fewer points of failure, and low cost.
Mesh, Client or AP mode Wifi	Can be a component in Mesh Wifi networks or connect to existing Wifi networks
One-IP configuration	Set one IP and the device is ready to make and receive phone calls. Quickly build voice and IP networks and roll out a Wireless PBX in minutes.
Web GUI or Phone UI	Configure via a web interface or telephone IVR menu. With Phone UI no laptop is required for on-site installation.
UV-resistant, weather-proof enclosure	Long life outdoors, no need to purchase a separate enclosure for outdoor use.
Built in mounting points	Simple and low cost outdoor installation.
Rugged design	Withstands abuse that would destroy other products such as power surges, brownouts, reverse DC, over-voltage DC, and static electricity.
Wide range DC power supply	Runs on any DC voltage from 10V to 40V, or any AC voltage from 110 to 250VAC via wall-plug type power supply. A solar panel can be directly connected – saving money on a solar regulator.
Power efficient	Consumes just 2.5W (DC). Can run on a 10W solar panel.
Power over Ethernet (PoE) and Power over Telephone Line (PoTL)	Just one cable run for both power and telephone. PoE and PoTL injectors included.
Open Hardware Design	No vendor lock-in. Open to improvement by anyone.
Open Source firmware	Linux, OpenWRT, B.A.T.M.A.N., and Asterisk. Stable, reliable community developed software. Open to improvement, adaptation, and innovation.
Asterisk	Extremely configurable, add IVR menus, connect IP Phones and billing systems

Overview

- Atheros AR2317 system on a Chip (SoC)
- Silicon labs FXS port chipset
- MIPS 4k processor 180 MHz
- One 10/100Mbit LAN port
- 8 MByte Serial Flash EEPROM
- 16 MByte RAM

Wireless LAN

- IEEE 802.11b/g
- Frequency Band : 2.4 to 2.462GHz
- Antenna Type : Internal Omnidirectional PCB Antenna
- Transmit EIRP power: 1-24 Mbit 20dBm or 36-54 Mbit 17dBm

Interfaces/Ports

- LAN Port : 1 x RJ-45
- FXS Port : 1 x RJ-11

Firmware

- Linux kernel 2.26.3
- OpenWRT Kamikaze (customized version based on version 8.09)
- B.A.T.M.A.N. mesh routing daemon Version 0.3
- Asterisk 1.4.11

Environmental

- Operating Humidity : ?
- Operating Temperature : ?

Electrical

- Power Type : wide range, protected DC port
- Power Options: AC adaptor (supplied) PoE or PoTL (Power over Telephone Line)
- DC Input Voltage : 9-40 VDC
- DC Input Power : 2.5W DC nominal
- Protected Ports: DC, RJ11 phone , RJ45 Ethernet
- Protection: overvoltage, reverse DC, nearby lightning, static electricity

Physical

- Casing: UV-protected, weatherproof outdoor enclosure
- LED Status Indicators : Power – LAN – etc
- Case Colour : Beige
- Width : 120mm
- Depth : 35mm
- Height : 220mm
- Weight : tba

Appendix B – Setting up automatic ssh into all mesh potatoes

There might be situations where you need to put or get files from all the meshpotatoes but you don't want to write the password accessing the MeshPotato for each time. This is how you deploy your public ssh key to the MeshPotatoes so you can ssh into it without getting prompted for a password.

Generate key on local machine

```
ssh-keygen -t dsa
```

It will ask you for a password but you can leave it blank.

Copy your local public key to the remote server

```
scp ~/.ssh/id_rsa.pub remote.server.com:/etc/dropbear/authorized_keys
```

Now ssh to the remote server

Now you can ssh to the remote server without entering your password.

Security

Now keep in mind that all someone needs to login to the remote server, is the file on your local machine `~/.ssh/id_dsa`, so make sure it is secure.

Making the script which ssh to all the MeshPotatoes

Do a ping, then SSH if it is open.

Acronyms

- B.A.T.M.A.N – Better Approach To Mesh Ad-hoc Network
- dBi: decibel isotropic, the antenna gain relative to an isotropic radiator (an antenna which radiates energy perfectly equal in all directions).
- LAN – Local Area Connection
- PoE: Power over ethernet (passive), power is transferred to a device through the unused twisted pairs inside a UTP cable.
- WLAN – Wireless Local Area Network

References

http://wiki.villagetelco.org/images/3/39/MP01_Hardware_Description_2010-05-02.pdf

<http://wiki.villagetelco.org>

<http://vtcamp.villagetelco.org>

