Database output for Freimap

Google Summer of Code 2010
WCW 2010 - Berlin
Who am I
Who am I

Name: Stefano Pilla
Who am I

Name: Stefano Pilla

Degree in Computer Science
Who am I

Name: Stefano Pilla

Degree in Computer Science

Contact: pillastefano@gmail.com
Who am I

Name: Stefano Pilla

Degree in Computer Science

Contact: pillastefano@gmail.com

http://wiki.freifunk.net/Freimap
Who am I

Name: Stefano Pilla
Degree in Computer Science

Contact: pillastefano@gmail.com

http://wiki.freifunk.net/Freimap
http://wiki.freifunk.net/FreimapWebApp
Who am I

Name: Stefano Pilla
Degree in Computer Science

Contact: pillastefano@gmail.com

http://wiki.freifunk.net/Freimap
http://wiki.freifunk.net/FreimapWebApp
http://wiki.ninux.org
What is Freimap?
What is Freimap?

Freimap is an Open Source visualization and analysis framework for (most) mesh networks, such as for example Freifunk.net. It can read many different data source and display them as different layers.
What is Freimap?

Freimap is an Open Source visualization and analysis framework for (most) mesh networks, such as for example Freifunk.net. It can read many different data source and display them as different layers.
What is Freimap?

Freimap is an Open Source visualization and analysis framework for (most) mesh networks, such as for example Freifunk.net. It can read many different data source and display them as different layers.
“First” Freimap
“First” Freimap

Coded by Thomas Hirsch
“First” Freimap

Coded by Thomas Hirsch

All displayed info are lost when you close the app
“First” Freimap

Coded by Thomas Hirsch

All displayed info are lost when you close the app

No direct interaction with nodes
“First” Freimap

Coded by Thomas Hirsch

All displayed info are lost when you close the app

No direct interaction with nodes

Is a Java based application
“First” Freimap

Coded by Thomas Hirsch

All displayed info are lost when you close the app

No direct interaction with nodes

Is a Java based application

Completely manually coded
“New” Freimap

GSoC2009/2010
“New” Freimap
GSoc2009/2010

Created with Netbeans IDE
“New” Freimap

GSoC2009/2010

Created with Netbeans IDE

Interaction with nodes
(SSH, ServiceDiscovery, SNMP)
“New” Freimap

GSoC2009/2010

Created with Netbeans IDE

Interaction with nodes (SSH, ServiceDiscovery, SNMP)

After GSoC2010 it will stores all information about nodes and links in a Database
“New” Freimap
GSoC2009/2010

Created with Netbeans IDE

Interaction with nodes
(SSH, ServiceDiscovery, SNMP)

After GSoC2010 it will stores all information about nodes and links in a Database

Micro-utilities: Search a node by lat/lon, filters, goToPosition, etc...
“New” Freimap

GSoC2009/2010

Created with Netbeans IDE

Interaction with nodes (SSH, ServiceDiscovery, SNMP)

After GSoC2010 it will stores all information about nodes and links in a Database

Micro-utilities: Search a node by lat/lon, filters, goToPosition, etc...
“New” Freimap

GSoc2009/2010

Created with Netbeans IDE

Interaction with nodes (SSH, ServiceDiscovery, SNMP)

After GSoc2010 it will stores all information about nodes and links in a Database

Micro-utilities: Search a node by lat/lon, filters, goToPosition, etc...
How it is structured
How it is structured

There are 3 main “Layers”
How it is structured

There are 3 main “Layers”

Datasource Listener
How it is structured

There are 3 main “Layers”

Datasource Listener

Central Layer
How it is structured

There are 3 main “Layers”

Datasource Listener

Central Layer

Main Layer
How it is structured

There are 3 main “Layers”

Datasource Listener

Central Layer

Main Layer

GSoC2010 project
Layer
Layer

The goal is to find a simple way to store only “Central Layer” data
Layer

The goal is to find a simple way to store only “Central Layer” data.

Other layers contain derivable information
  (ie. MainLayer converts LatLon in XY position)
Layer

The goal is to find a simple way to store only “Central Layer” data.

Other layers contain derivable information (ie. MainLayer converts LatLon in XY position).

Create a method to quickly find these data in the database and draw it on the map.
How Database is structured
How Database is structured

Fields and tables depend on which wireless communities you are connected.
How Database is structured

Fields and tables depends on which wireless communities you are connected

i.e Main tables could be:
How Database is structured

Fields and tables depends on which wireless communities you are connected

i.e Main tables could be:

Layer
How Database is structured

Fields and tables depends on which wireless communities you are connected

i.e Main tables could be:

Layer

Links
How Database is structured

Fields and tables depends on which wireless communities you are connected

i.e Main tables could be:

Layer

Links

Nodes
How Database is structured

Fields and tables depends on which wireless communities you are connected

i.e Main tables could be:

- Layer
- Links
- Nodes
- Interfaces
Nodes Table

FQID - IP - Lat - Lon - isGw - GwIP - Uptime - (etc..)
<table>
<thead>
<tr>
<th>FQID</th>
<th>IP</th>
<th>Lat</th>
<th>Lon</th>
<th>isGw</th>
<th>GwIP</th>
<th>Uptime</th>
<th>(etc..)</th>
</tr>
</thead>
</table>

Nodes Table
## Nodes Table

<table>
<thead>
<tr>
<th>FQID</th>
<th>IP</th>
<th>Lat</th>
<th>Lon</th>
<th>isGw</th>
<th>GwIP</th>
<th>Uptime</th>
<th>(etc..)</th>
</tr>
</thead>
</table>

Interfaces record
Nodes Table

FQID - IP - Lat - Lon - isGw - GwIP - Uptime - (etc..)

Interfaces record

(NodeID, MainAddress(bool), ip)
Nodes Table

FQID - IP - Lat - Lon - isGw - GwIP - Uptime - (etc..)

Interfaces record

(NodeID, MainAddress(bool), ip)
Nodes Table

FQID - IP - Lat - Lon - isGw - GwIP - Uptime - (etc..)

Interfaces record booleay

(NodeID, MainAddress(bool), ip)
<table>
<thead>
<tr>
<th>ID</th>
<th>NodeS</th>
<th>NodeD</th>
<th>FirstTimeSt</th>
<th>LastTimeSt</th>
<th>....</th>
</tr>
</thead>
</table>
Layer Table

ID - NodeS - NodeD - FirstTimeSt - LastTimeSt - ....

are a NodeID identify a Link
Layer Table

ID - NodeS - NodeD - FirstTimeSt - LastTimeSt - ....

are a NodeID identify a Link

If two or more layers have the same information I update only LastTimeSt
Layer Table

**ID** - NodeS - NodeD - FirstTimeSt - LastTimeSt - ....

are a NodeID identify a Link

If two or more layers have the same information I update only LastTimeSt

In this way if the network does not change, then for a period of time I can use a single record only
Links Table
<table>
<thead>
<tr>
<th>Src</th>
<th>Dest</th>
<th>FirstTimeSt</th>
<th>LastTimeSt</th>
<th>Etx</th>
<th>Lq</th>
<th>nlq</th>
</tr>
</thead>
</table>

Links Table

Src - Dest - FirstTimeSt - LastTimeSt - Etx - Lq - nlq

is a NodeID
<table>
<thead>
<tr>
<th>Src</th>
<th>Dest</th>
<th>FirstTimeSt</th>
<th>LastTimeSt</th>
<th>Etx</th>
<th>Lq</th>
<th>nIq</th>
</tr>
</thead>
</table>

is a NodeID

This info are in sync with the routing protocol updates
Freimap Requirements
Freimap Requirements

MySql Server
Freimap Requirements

MySQL Server

OLSR Network
Freimap Requirements

MySql Server

OLSR Network

Java Environment
Freimap Requirements

MySQL Server ↔

OLSR Network ↔

Java Environment ↔
Freimap Requirements

MySQL Server  <->  People won’t install a DB

OLSR Network  <->

Java Environment  <->
Freemap Requirements

MySql Server <-> People won’t install a DB

OLSR Network <-> If I’m not connected to an OLSR network?

Java Environment
Freimap Requirements

MySQL Server  People won't install a DB

OLSR Network  If I'm not connected to an OLSR network?

Java Environment  Bad Performance but high compatibility
Freemap Requirements

MySql Server  People won’t install a DB

OLSR Network  If I’m not connected to an OLSR network?

Java Enviroment  Bad Performance but high compatibility

Solution: Port Freimap as a javascript WebApp in a central Server
Freimap WebApp - Conclusions

1. Step - Define a common node database schema
Freimap WebApp - Conclusions

I Step - Define a common node database schema

This means interoperability of all wireless communities
Freimap WebApp - Conclusions

1 Step - Define a common node database schema

This means interoperability of all wireless communities

THIS IS REALLY IMPORTANT!
Freimap WebApp - Conclusions

I Step - Define a common node database schema

This means interoperability of all wireless communities

THIS IS REALLY IMPORTANT!

II Step - Make a porting from Java to JavaScript
Freimap WebApp - Conclusions

I Step - Define a common node database schema

This means interoperability of all wireless communities

THIS IS REALLY IMPORTANT!

II Step - Make a porting from Java to JavaScript

Porting from Java to Javascript is to difficult to apply
Freimap WebApp - Conclusions

**I Step** - Define a common node database schema

This means interoperability of all wireless communities

**THIS IS REALLY IMPORTANT!**

**II Step** - Make a porting from Java to JavaScript

*Porting from Java to Javascript is to difficult to apply*

**III Step** - Create a package for basic installation of the WebApp
Freimap WebApp - Conclusions

I Step - Define a common node database schema
This means interoperability of all wireless communities
THIS IS REALLY IMPORTANT!

II Step - Make a porting from Java to JavaScript
Porting from Java to Javascript is too difficult to apply

III Step - Create a package for basic installation of the WebApp
In this way the WabApp could be used by all Wireless Communities
Suggestions?
Suggestions?

Q&A
Service Discovery in Freimap

Google Summer of Code 2009
Freimap
Gsoc project
Main goal was to create a datasource for Service Discovery in Freimap
Freimap
Gsoc project

Main goal was to create a datasource for Service Discovery in Freimap

Bonjour/Avahi is a suite of protocols that consists of 3 services:
Freimap

Gsoc project

Main goal was to create a datasource for Service Discovery in Freimap

Bonjour/Avahi is a suite of protocols that consists of 3 services:

IPv4LL (IPv4 Link Local Addressing)
Freimap
Gsoc project

Main goal was to create a datasource for Service Discovery in Freimap

Bonjour/Avahi is a suite of protocols that consists of 3 services:

IPv4LL (IPv4 Link Local Addressing)
For automatic IP configuration in a network
Bonjour/Avahi is a suite of protocols that consists of 3 services:

- **IPv4LL (IPv4 Link Local Addressing)**
  
  *For automatic IP configuration in a network*

- **mDNS (multicast DNS)**

Main goal was to create a datasource for Service Discovery in Freimap.
Bonjour/Avahi is a suite of protocols that consists of 3 services:

IPv4LL (IPv4 Link Local Addressing)  
For automatic IP configuration in a network

mDNS (multicast DNS)  
For IP to Hostname (e viceversa) translation in a distributed networks

Main goal was to create a datasource for Service Discovery in Freimap
Freimap
Gsoc project

Main goal was to create a datasource for Service Discovery in Freimap

Bonjour/Avahi is a suite of protocols that consists of 3 services:

IPv4LL (IPv4 Link Local Addressing)
For automatic IP configuration in a network

mDNS (multicast DNS)
For IP to Hostname (e viceversa) translation in a distributed networks

DNS-SD (DNS - Service Discovery)
Bonjour/Avahi is a suite of protocols that consists of 3 services:

- IPv4LL (IPv4 Link Local Addressing)
  - For automatic IP configuration in a network
- mDNS (multicast DNS)
  - For IP to Hostname (and vice versa) translation in a distributed network
- DNS-SD (DNS - Service Discovery)
  - Service Discovery with DNS query

Main goal was to create a datasource for Service Discovery in Freimap.
Freimap
Service Discovery con mDNS
Freimap

Service Discovery con mDNS

The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc..)
Freimap

Service Discovery con mDNS

The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc..)

For example if I want to discover all printers on the network i can make this query: _ipp._tcp.local.
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc.).

For example if I want to discover all printers on the network i can make this query: _ipp._tcp.local.

An mDNS query:
Freimap
Service Discovery con mDNS

The main goal is to discover all possible services on the mesh network (i.e., WebServer, Skype User, a printer, etc.).

For example if I want to discover all printers on the network I can make this query: _ipp._tcp.local.

An mDNS query:
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc.).

For example if I want to discover all printers on the network I can make this query: _ipp._tcp.local.

An mDNS query:
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc..)

For example if I want to discover all printers on the network i can make this query: 

```
_ipp._tcp.local.
```
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc.).

For example if I want to discover all printers on the network i can make this query: _ipp._tcp.local.
The main goal is to discover all possible services on the mesh network (i.e., WebServer, Skype User, a printer, etc.).

For example, if I want to discover all printers on the network, I can make this query: _ipp._tcp.local.

An mDNS query:
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc..)

For example if I want to discover all printers on the network i can make this query: _ipp._tcp.local.
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc.).

For example if I want to discover all printers on the network i can make this query: _ipp._tcp.local.

An mDNS query:
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc.).

For example, if I want to discover all printers on the network I can make this query: _ipp._tcp.local.

An mDNS query:
The main goal is to discover all possible services on the mesh network (i.e. WebServer, Skype User, a printer, etc.).

For example, if I want to discover all printers on the network, I can make this query: `_ipp._tcp.local`.

An mDNS query:
Freimap

Service Discovery in Freimap
In the “new” Freimap there are two kind of query:
Freimap

Service Discovery in Freimap

In the “new” Freimap there are two kinds of query:

Multicast Query
In the “new” Freimap there are two kind of query:

Multicast Query

Unicast Query
In the “new” Freimap there are two kind of query:

**Multicast Query**

**Unicast Query**

**Multicast Query:**
In the “new” Freimap there are two kind of query:

Multicast Query

Unicast Query

Multicast Query:
Address: 224.0.0.251
Freimap
Service Discovery in Freimap

In the “new” Freimap there are two kind of query:

Multicast Query
Unicast Query

Multicast Query:
Address: 224.0.0.251

Discover all service on the network
In the “new” Freimap there are two kind of query:

Multicast Query

Unicast Query

Multicast Query:

Address: 224.0.0.251

Discover all service on the network

Answer could be to the multicast address
In the “new” Freimap there are two kinds of query:

**Multicast Query**

**Address:** 224.0.0.251

Discover all services on the network

Answer could be to the multicast address

**Unicast Query**
Freimap
Service Discovery in Freimap

In the “new” Freimap there are two kind of query:

Multicast Query

Unicast Query

Multicast Query:
Address: 224.0.0.251
Discover all service on the network
Answer could be to the multicast address

Unicast Query:
Address: Main address of the node
Freimap
Service Discovery in Freimap

In the “new” Freimap there are two kind of query:

**Multicast Query**

Address: 224.0.0.251
Discover all service on the network
Answer could be to the multicast address

**Unicast Query**

Address: Main address of the node
Discover only services of one node
Freimap
Service Discovery in Freimap

In the “new” Freimap there are two kind of query:

Multicast Query

Unicast Query

Multicast Query:
Address: 224.0.0.251
Discover all service on the network
Answer could be to the multicast address

Unicast Query:
Address: Main address of the node
Discover only services of one node
Answer to the originator of the request
About Node

Node Name: StefanoP
Lat/Lon: 41.8637595/12.5535823
UpTime: 4 days 3h 2m
Attributes: Gateway - 192.168.1.1

IP: 192.168.1.103
Plugins: mDNS

SNMP Graphs

Service Discovery

Type: _http_.tcp.local.
_skype_.tcp.local.
_daap_.tcp.local.
_presetence_.tcp.local.

Services:
Libreria di Stefano Pilla

Detail:
Libreria di Stefano Pilla
MacBook-Pro-di-Stefano-Pilla.local.:3689
192.168.1.103:3689
Version=196616
Database ID=A6745260B09E13F4
txtvers=1
Machine ID=E93E3D9DEA87
Conclusion
Conclusion

Freimap is a really potential software but there are some things that must be improved.

Service Discovery and Database output is only a part of Freimap.

Weakness:
Conclusion

Freimap is a really potential software but there are some things that must be improved.

Service Discovery and Database output is only a part of Freimap.

Weakness:

Performance
Conclusion

Freimap is a really potential software but there are some things that must be improved.

Service Discovery and Database output is only a part of Freimap.

Weakness:

Performance

Scalability
Conclusion

Freimap is a really potential software but there are some things that must be improved.

Service Discovery and Database output is only a part of Freimap.

Weakness:

Performance

Scalability

Use
Conclusion

Freimap is a really potential software but there are some things that must be improved.

Service Discovery and Database output is only a part of Freimap.

Weaknesses:

Performance
Scalability
Use

Solution: Porting Java Freimap as a WebService with a PHP Framework
Info/Updates
Info/Updates

http://wiki.freifunk.net/Freimap
Info/Updates

http://wiki.freifunk.net/Freimap

http://wiki.freifunk.net/FreimapWebApp
Info/Updates

http://wiki.freifunk.net/Freimap

http://wiki.freifunk.net/FreimapWebApp

http://wiki.ninux.org/GSoC
Info/Updates

http://wiki.freifunk.net/Freimap

http://wiki.freifunk.net/FreimapWebApp

http://wiki.ninux.org/GSoC

We waiting you in Rome at the WBMv3 / 2 - 6 June 2010
Info/Updates

http://wiki.freifunk.net/Freimap

http://wiki.freifunk.net/FreimapWebApp

http://wiki.ninux.org/GSoC

We waiting you in Rome at the WBMv3 / 2 - 6 June 2010

More info: http://battlemesh.org - contatti@ninux.org
Q&A