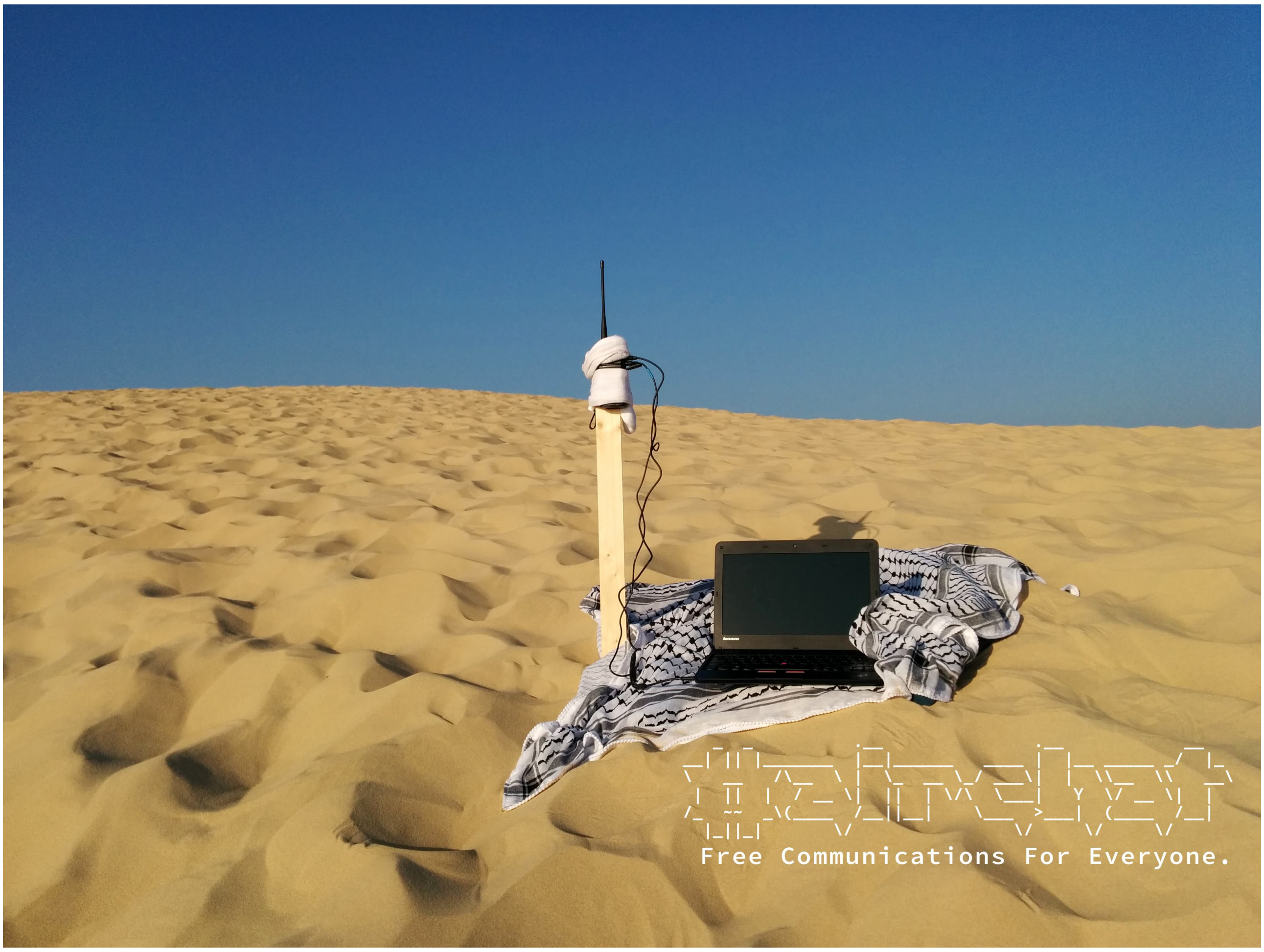
AirChat

all your air waves are belong to us



*(off-road warrior epic mobile station)

Communications as the Air itself must be free.

#lulzlabs
Anonymous

Why Airchat?

Because we strongly believe communications should be free,
Free as much as the air itself and all the waves should be.
Free for everyone everywhere, free for those oppressed, free for the poor,
free for the dissident, free for those living out of the boundaries
of the infrastructure created for those who were lucky enough to have more than others.
And free...well... because sometimes the non-free infrastructure itself fails.

Several thousands years ago, we started shouting into the air to
communicate, to build our first communities and to survive.
Since then the power of our voices has travelled through the air,
carrying poetry, intelligence, knowledge, art, emotions, science,
revolutions, philosophy, faith, evil, war,
Transmitting all those ideas, good or bad, which define us as human.

We freely shouted to the air our very own existence.
We shouted to the wind we were alive.

Several thousands of years after we started this adventure, we have built amazing and
technologically sophisticated networks to serve us to communicate everywhere.
Now the fire of our freedom is burning away.
Our voices, once free, are subject to uncountable controls, financial fees, patents, rights,
regulations, government censorship, etc.

Today, we have acknowledged that, even after all these years of technology advance,
we still need to meet in common public places to continue expressing ourselves in a free way,
to build up our sense of community and stand up for our future and rights.
Our so advanced communication infrastructure has failed to make us a better family, to make us
better humans, to bring us openness, democratic access and freedom to think nor to speak.
Our pay-to-participate infrastructure identifies us, targets us, monitors us, controls us.
so then, we will try to go to the origin of all and try to scale up all these really very human voices
to cover not only those tiny public spaces but a whole community, a neighbourhood,
a big town, a huge city, a remote region... the world.

So...That's why Airchat,
because next time you want shout your freedom to the wind, perhaps someone will hear you.

WTF is AirChat then?!?!

Airchat is a free communication tool,
free as in 'free beer'
and free as in 'Jeremy Hammond must be freed'.
It doesn't need the internet infrastructure, nor does it need a cellphone network,
instead it relies on any available radio link (or any device capable of transmitting audio -
we even made a prototype working with light/laser based transmissions).

This project was conceived not only from our lessons learned in the Egyptian, Libyan and Syrian revolutions, but also from the experience of OccupyWallStreet and Plaza del Sol.
We have considered the availability of extremely cheap modern radio devices (like those handhelds produced in China), to start thinking about new ways in which people can free themselves from expensive, commercial, government controlled, under surveillance infrastructure.

AirChat is not only our modest draft or proposal for such a dream,
but it is a working PoC you can use today.
We hope you will enjoy it and we also hope that you too will be able to feel the beauty of free communications,
free communications as in 'free beer'
and free communications as in 'free yourself and your people forever'.

User cases

People who were protesting against their govt. resulting in the their internet being cut off.
Even worse govt. decided to fuck with their cellphones networks too.
They need basic communication tools to spread news and updates about their conditions,
and with the aim to eventually relay that information to/from the internet when at least one of them is able to get a working internet connection.

NGOs and medical teams working in Africa under poor conditions who want to build some basic communication's infrastructure to coordinate efforts like the delivery of medication and food or to update on local conditions without being intercepted by regional armed groups etc.

Dissident groups who mistrust the normal communication infrastructure and who want to coordinate regional activity and share updates about oppressive actions carried out by the authorities.

Disaster response, rescue and medical teams who are working in devastated zones without the availability of standard telecommunication infrastructure. They want to keep updating their statuses, progress and resource availability between teams when there may be large overage zones between them.

Yacht owners who are sailing and who wish to obtain news updates from some approaching coastline or another ship which has internet access. There may just be a simple exchange of information about news, weather conditions, provisions, gear etc.

Local populations who want to keep in touch with each other on a daily basis with the goal of developing a strong community capable of maximizing their resources, food or manpower to help improve sustainability and their quality of life.

Street protests or any other street event where people would like to share their thoughts, anonymously and locally without relying on the internet. They may also wish to share them with the world as a single voice using a simple gateway such as a unique Twitter account made for the occasion.

Expedition basecamps who need a simplistic solution to build a common gateway for establishing radio communication and messaging service links with camps, remotely located basecamps and/or rescue teams to coordinate tasks such as logistics, rescue efforts, routes and schedules.

Background: Dilemmas and decisions

Every project is a fractalized representation of infinite dilemmas sparkling other new ones,
glued to the futile decisions we make, to try to address all of them.

Many ideas have crossed our minds when we tried to make this thingie.
We experimented broadcasting UDP packets inside mesh network solutions.

We experimented using patched wireless network card drivers to inject crafted wifi management frames.
We also considered crafting TDMA packets via cellphones RF hardware.
We thought about those many different possibilities. we saw there's so much potential on them.
Sadly we found out how locked down and over-regulated our communication devices are.

So, we thought: 'Well some solutions would require that we ask people to root their phones or routers, and to then install custom firmwares with patched drivers, with the risk of getting people mad cause they were bricking them'.

we also thought about a Wifi interconnected cellphone net approach, but the coverage range was rather frustrating.

We saw people working on different mesh related projects and we thought
'One solution shouldn't discard another one but it should try to complement it,
to add interoperability and to allow heterogeneous systems'.
As different serious projects are looking for solutions based on 802.11 standards we said,
'WTF lets try to reinvent the wheel for exploration and fun'.

But to reinventing the wheel you need freedom. a freedom which we don't have much of in our world of telecommunications, which is over-regulated by evil organizations like the FCC and similar shits around the world.
So we choose a good ol' trusted ancient technology to start free.

Radio transceivers.
yeah, these shits rock.
we chose to sacrifice bandwidth for freedom.
Tune the frequency.
Define a protocol.
Transmit.
Enjoy.

So yeah we connected our 897Ds to our computers, we shouted out to our bros to tune in and then we started playing around.

Initially AirChat used code from 'minimodem' and then from 'soundmodem' sources but after suggestions from the ham radio people involved in ARES, we decided to separate it and to make it modular to use Fldigi software, a broadly deployed solution for use with ham radios.
Fldigi is controlled by means of XML-RPC calls which can be made even between remote systems (example: One workstation, connected to radio equipments is dedicated to listening to the radio frequencies constantly while another system is running AirChat etc.)
We are open to feedback about this decision, and we will offer different implementations if they're needed.

We ended up with a simple protocol packet: the Lulzpacket. This simple packet contains information to verify there was no corruption during the transmission and a random code to pseudo-identify the packet. We define the addresses of nodes in the net by their ability to decrypt a given packet. Addresses are derived from the hashes of asymmetric encryption keys, Every radio node defines its own address by the pair of keys it has generated for itself and the addresses change if users choose to regenerate their keys. Each node only cares for what is being received. No hardware identification, no transmitter plain identification. only packets matter. transmissions are anonymous. whenever an address is needed to reply to a packet, it is encrypted inside the packet.
Packets targeting specific addresses are encrypted and they must be decrypted by the private key only the target possesses. Anyone trying to spoof an address will not be able to decrypt the packet.

Symmetrical encrypted packets are available also and can be used as an extra
layer too. General non-encrypted packets also available by default for general
broadcasting and community discussion. (also for those people on some countries
where laws forbid encryption on certain radio frequencies, etc).
Disclaimer:
we don't give a fucking shit about prohibitions over the use of encryption. fuck you NSA.

So the choice is yours. You can use it with encryption or without.
Encryption is part of the routing solution approach, non encrypted packets are linked
to general broadcast.
(remember that when you are in the middle of a massive crisis you probably wouldn't care much
about the stupid FCC)

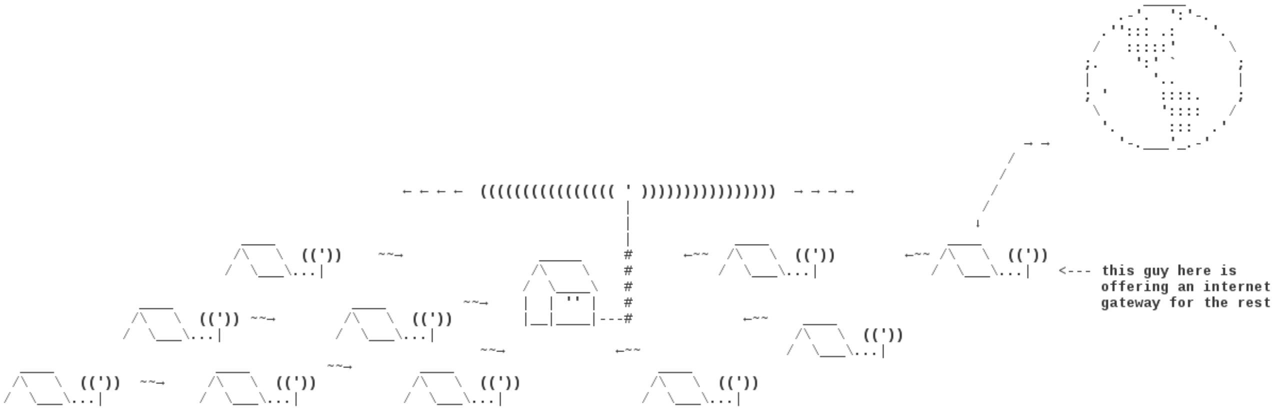
Airchat is our first service which implements this protocol. The current release right now focuses on messaging and it can be used as a simplistic message board inside a LAN and to rely communications between radio nodes. It has built-in internet gateway capabilities to offer users access to some basics such as tweeting, retrieving twitter streams, downloading news, community related articles, etc.
This gateway can can be used whenever an Airchat running station gains a working internet connection and choose to share it.
(this internet access can be anonymized via Tor and the built in proxy support).

The first release will be a minimal set of useful functionality, so that we can see what people can do with it and what they would like to be able to do. We will continue to add more features based on your feedback.

So far we have played interactive chess games with people at 180 miles away. we have shared pictures and established encrypted low bandwidth digital voice chats. We have 3D printed over distances of 80 miles and transmitted medical orders at distances of over 100 miles.
All without phones or internet access.

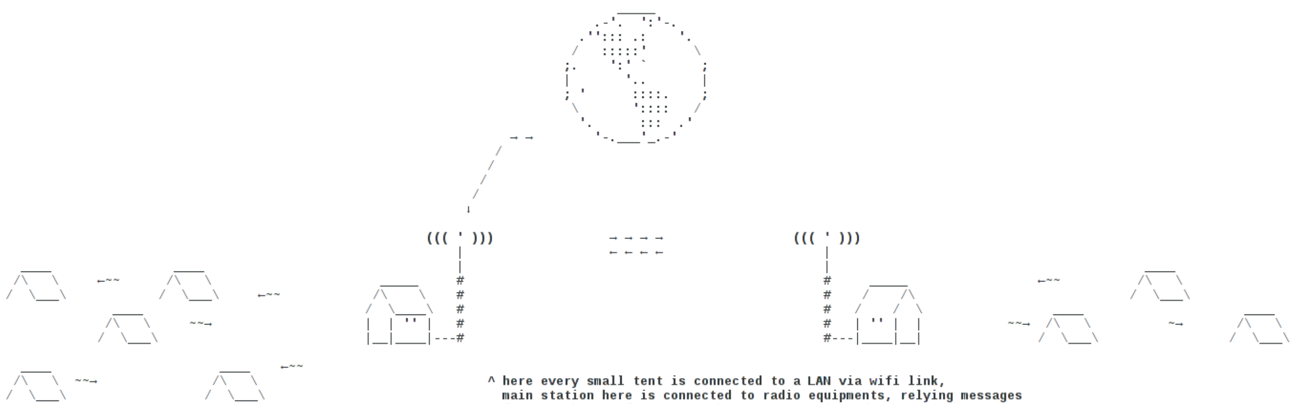
So how does it feel when you are communicating freely?
it feels great...fucking great.

Some Possible RF Network Configurations



<--- this guy here is
offering an internet
gateway for the rest

^ everyone is sending messages to a main station (pirate fm station) which is broadcasting all of them together,
uplink uses different frequencies than the main station uses for streaming on the downlink



```
^ here every small tent is connected to a LAN via wifi link,
  main station here is connected to radio equipments, relying messages
  from/to the LAN from/to another remote radio node.
  (optionally offering an internet gateway to everyone if it's available)
```

Quick Start

Software Setup

FreeBSD 10
from a fresh server install:

```
# pkg install make
# pkg install perl-5.16.xx
# perl install-modules-airchat-freebsd.pl
```

then...

```
# cpanp
CPAN Terminal> i Net::Server --skiptest
CPAN Terminal> i HTTP::Server::Simple::CGI::PreFork
```

that will get you the airchat server running,
keep in mind installing fldigi requires a graphical environment aka X
so,
1: you setup airchat to connect with a remote station running fldigi
2: install a graphical environment and then:

```
# pkg install fldigi-3.xx.xx
```

Windows
Install Strawberry Perl >= 5.18 (the portable zip version fits well for example)
from http://strawberryperl.com/

Direct link:
http://strawberryperl.com/download/

Once you get perl installed, run in your perl shell:

```
# perl install-modules-airchat-windows.pl
```

Then install these modules via the cpanplus terminal:

```
# cpanp
CPAN Terminal> i Net::Server --skiptest
CPAN Terminal> i HTTP::Server::Simple::CGI::PreFork –skiptest
```

Then install fldigi from:
http://www.w1hkj.com/download.html

Linux (Debian / tested also on Ubuntu Trusty)
Install some needed stuff:

```
# apt-get install make libcpanplus-perl libhttp-server-simple-perl libcrypt-cbc-perl
libcrypt-rijndael-perl librpc-xml-perl libxml-feedpp-perl liblwp-protocol-socks-perl
libnet-twitter-lite-perl libnet-server-perl
```

(There's an optional and commented "use Net::SSLGlue::LWP" before "use LWP::UserAgent" on airchat.pl (# apt-get install libnet-sslglue-perl),
This magically fixes LWP for https requests issues, when, for example, you want to include feeds only available via proxy to a https address, if you don't have the updated libwww-perl 6.05-2 and liblwp-protocol-https-perl 6.04-2 available from repositories (should be available from the jessie repos thou)) but...We strongly recommend you look to update libwww-perl and liblwp-protocol-https-perl to their latest versions, cause using SSLGlue will eventually break https access to the twitter API.
Check if you have updated packages for 'libnet-twitter-lite-perl' because you will need the Twitter API v1.1 support for the built-in Twitter Gateway.
run:

```
# perl install-airchat-modules-linux.pl
```

^ this will install 'HTTP::Server::Simple::CGI::PreFork' (needed) and 'Net::Twitter::Lite::WithAPIv1_1'

If you want to install Fldigi on the same machine than Airchat then:

```
# apt-get install fldigi
```

(running fldigi requires a graphical environment)

MacOS X
Get XCode.
Launch XCode and bring up the Preferences panel. Click on the Downloads tab.
Click to install the Command Line Tools. **Check you got 'make' installed.**
run:

```
# perl install-airchat-modules-macosx.pl
# cpanp
CPAN Terminal> i Net::Server --skiptest
CPAN Terminal> i HTTP::Server::Simple::CGI::PreFork --skiptest
```

General Notes

Airchat runs by default on port 8080, connect your browser to (for Example: http://localhost:8080). READ THE CODE.
If you find some problem running airchat, please try updating modules and linked libraries.
we've found some issues related to outdated implementations.
(like '500 Bad arg length for Socket6::unpack_sockaddr_in6, length is 16, should be 28' happening in Ubuntu Precise when enabling the Twitter gateway)

Fldigi Setup

run fldigi.
skip everything if you want but you must configure audio devices to make it work with your capture device and your audio output device. test if it's working capturing audio signals and playing audio. and that's all.

(Note: keep your fldigi updated always)

Hardware Setup

Radio transceivers usually come with many different interfaces,
Each brand deploys different connectors even within their own range of models and
sadly there's usually no standard which they follow.

We understand that some people have experience using more expensive
radio equipments and will know how to link those transceivers to their computers.
As such we will focus on supporting the cheapest and most accessible models which are able
to offer the democratization of this solution worldwide even in the poorest regions.

We have considered cheap Chinese vhf/uhf fm handheld transceivers
available worldwide at as low as $40 bucks each.

These devices come with a Kenwood 2-pin connector composed by a 2.5mm jack and a 3.5mm one.
The 2.5mm jack transports the speaker signal and the 3.5mm serves as the microphone input.

We will make a very simple setup using the VOX function on the transceiver to avoid more complex PTT setups.

First connect some 2.5mm male to 3.5mm male cable between the speaker output on the radio and the microphone input on your computer.

Then take a stereo 3.5mm male to 3.5mm male cable and cut all of the small cables inside except the red one (It should be a red cable which is connected to the middle ring of the jack).

Only the red cable with the signal coming from the ring of the 3.5mm jack should be connected and nothing else.
(neither the tip, nor the ground (ground will be provided by the 2.5mm jack cable)).

Once you are done, connect this customised cable to the microphone input on the
radio transceiver and then to the speaker output of your computer.

Finally, set the frequency everyone has agreed to use on the transceiver,
Don't forget to enable the VOX function (adjust the sensitivity to medium).
Modify the transmission timer to more than 2 minutes, set the radio speaker volume to approx. 50%, tune the microphone sensitivity on your computer to base levels with medium boost (if needed) and finaly set the computer headphones volume to around 75% or so and then you are ready to go. Keep testing till getting the best audio quality for your transmission.

Be careful about the quality of cables and soldering used, test the audio quality until getting the most optimal conditions possible, that will directly improve your transmissions.

Some Questions...

Audio transmission?
Almost every single home in this world has a common AM and/or FM radio.
In such cases when not everyone is able to get some cheap radio transceiver,
everyone at least will be able to decode packets being transmitted via a pirate FM stations (or AM) AM doesn't suffer the capture effect of FM. so under certain circumstances people could accommodate around 18 or 20 parallel different packet transmissions on the same bandwidth used for voice transmissions. Also it turned out to be cheap and simple to link laptops and radios via the soundcard, simple enough to allow easy-to-make off-road warrior RF enabled mobile stations.

Bandwidth?

We traded bandwidth for freedom, or to be more exact we traded bandwidth for freedom, simplicity and low cost, which indeed are the real conditions needed to democratize this solution. so yeah. sorry about the bandwidth but we do not regret it. We will be looking for solutions to this in the future but keep in mind that 'freedom, simplicity and low cost' won't be given up.

Is 4K video streaming coming soon?

no, like...no.

and pics?

We love pix. pix support is coming ofc. We have tested image transfers using Google's WebP format to try conserve bandwidth as much as possible, but the lack of support in several browsers has given us second thoughts. We will looking for further feedback about it.

I want to cyber my girlfriend (who lives 20 miles away) without having NSA agents fapping to it, can I use this for it?

ofc, man. thou we require your girlfriend to deliver tits or gtfo. (sorry but it's needed to help us on the datamining of frequency usage and transmission modes performance raw data through our Hadoop cluster of ARM servers, all those pix will be used for the datalink test..
err...hmm...err...derp)

What happened to Sabu?

He ended up working as a male prostitute for FBI, he usually wears a pinkish silk kimono in the evenings and he resides in Chattanooga.
He recently betrayed his male prostitute co-workers cause he was jelly they were getting more attention than him.
His sentencing is still being delayed by FBI.

Should Molly get on Jabber?

Yes.

FBI has been going after our bittie$. If you want to help, drop some penniez here:

1Kx4wVYBvbL6khNhA3SmJKnT8ZLeJHPBxA

^ bitcoins

#lulzlabs

2014. Anonymous. All Your Base Are Belong To Us.