Baculus

A resiliant, self-contained, friendly internet buoy

Team members

Jonathan Dahan - hi@jonathan.is - system architect, programmer

Ariel Cotton - ariel@argoncobalt.com - designer, programmer

Problem Statement

For coordinators / first responders, repairing a network is tricky business. Fixed location infrastructure is more efficient, but when it goes down, figuring out where temporary networks need to be placed to connect to the people that need it most is tricky.

For the people caught in a disaster, communications infrastructure is crucial for finding out if loved ones are safe, and finding out what resources (food, warmth, medical attention) are available where.

Solution Statement

We are talking a transparent plastic backpack with wheels, stuffing it with hardware and software that helps community leaders know where they should be moved to connect to the maximum area, and can run independent of any other infrastructure for up to 72 hours.

This backpack contains all the necessary (off-the-shelf) hardware and open source software to provide for cell phone and wifi connectivity, along with a map-view and message board view of messages.

It is meant to be used year-round as a public wifi access point.

Users

There are two users baculus helps - those who are in need after a disaster, and those who help coordinate before and during. The system is designed to be installed with community leaders in high risk areas, for the community to become familiar with the utility of a local-first message board system. This benefits users by allowing them to define and share what is important.

A key feature of the Red Hook Wifi initiative is that people were using the editable map and other local-first applications well before Sandy. This made it a natural response to use during a disaster.

Community leaders need help in disemminating information to those who might not know - where lunch can be had; and people have to share needs - if they are limited in mobility and need warmth, or medicine, or just to checkin that they are safe. Baculus provides this by allowing SMS broadcast and response, that is hooked up to a message board view for responders to coordinate.

Community & Location

The ideal locations for baculus are in medium sized communities around 35,000 people, that are at a high risk of natural disaster. The system is designed to scale down to small communities of around 250 and up to 100,000 people, but certain features like non-satellite beacon finding work best when the are has less dense buildings, for instance.

Communities that have limited (either slow or non-existant) internet access also benefit the most, by having baculus be immediately useful in non-disaster scenarios as a local network with applications that work fine disconnected from the greater internet, or even with intermittent power.

For example, many high-population growth urban areas have rolling brownouts, which can affect greater internet connectivity. Baculus would work fine in such a scenario, allowing for text and local applications to sync opportunisticly.

Technical Feasability

> *Explain the technical design of your solution. What technical capabilities do you envision a working prototype will have? Are there any technical hurdles or problems that will need to be solved in order to build a working prototype? At what stage of development is your solution currently?*

Design Philosophy

All the technologies, and design have been chosen with the following values in mind - **transparency & community friendliness**, **availability & affordability of hardware**, **documentation**, and **building platforms over products**. With this in mind, we have also decided to go with hardware and software we have already tested and worked, but with the goal that each could be replaced by more readily available pieces while still leveraging the rest of the system.

Hardware

The basic hardware design is a lithium ion battery powering a raspberry pi, coordinating a limeSDR, RTL-SDR, and ubiquiti litebeam access point. This will create three networks

1. Iridium network reciever for unconnected nodes to recieve GPS coords
2. Wifi mesh network for screen-based devices to connect to the map and message board applications for coordinating people and places
3. SMS network for feature phones to share needs and find friends and family

The most complicated component of baculus is communicating with the Iridium satellite network. This has been shown to work with an RT-SDR and HackRF in this talk. Independently, we have been able to tune into Iridium with our local RT-SDR but still need to work on making it robust and writing the software to send and decode GPS data.

For running the SMS interface, we have already build openBTS cellular networks, and had successfully gotten over 40 cell phones to autoconnect, receive broadcasts from a locally running web application, and send SMSs back during arthackday link here.

Network & Operating System Software

For the WiFi network, we will be using QMP images, the same used to power the over 34,000 nodes in Guifi and NYC Mesh. Any modifications / setup that cannot be upstreamed as options will be published as open source build scripts.

For the Iridium interface, we will be building on top of gnuradio - any new blocks will be published and available.

The cell phone network will run on OsmoBTS, which supports the LimeSDR Mini, whose developers have shown a working GSM networking on the Raspberry Pi 3. Having experience writing blocks for PiBakery , we will publish reproduceable builds on top of Raspbian (and upstream any blocks we can).

Applications

We will first test forum software like patchwork, before developing our own custom solution for message boards - it is end-to-end encrypted, offline-first, and syncs opportunisticly. Evaluation of building new features like the map view on top of patchbay will also be considered.

Any new software, like the SMS gateway, or map view, will use scuttlebutt to sync data, and be written in javascript on top of node.js to be approachable to the widest range of potential contributors as possible.

We have experience writing offline first open source web applications, from small prototypes that use no libraries or frameworks like this one-file schedule sketching application that can share state in the url to synchronized collaborative wiki editing link here. We will err on the side of using plain old javascript, and target the default Android 4.0 browser and Firefox 52 for the first prototype of frontend web applications.

For directing where the backpack should be moved to, we are building a divining rod that lights up, vibrates, and shows how many minutes they should walk in that direction. It will be powered by the raspberry pi, and a non-digital compass will be embedded in case the electronics fail.

Differentiation

> *How does the proposed solution differ from or improve upon existing solutions? What is innovative or novel about the proposed concept or technology?*

The main differentiator is that the nodes know their physical location on earth and help share where they should be placed by the humans. In addition, the closest comparable open source systems are about 3x as expensive (see LimeNet Mini @ $2400), and are based on a centralized structure.

Affordability

All parts chosen contain necessary power adapters, storage, antennae, etc.

PART	FUNCTION	COST
Charge Controller	USB Power, Solar adapter	$20
Litebeam 5AC gen2	WiFi 4km range @ 5m height	$75
Raspberry Pi 3	OS and application host	$60
LimeSDR Mini	GSM/LTE base station + wayfinding	$180
Clear Rolling Backpack	Case	$40
RTL-SDR	Iridium Reciever	$25
12V 35Ah Lithium Battery	Lightweight long-lasting battery	$290
30W Solar panel	Where appropriate	$50
	Total	**$700**/$750

The majority cost is $300 for a battery that is only 6.6lbs and rated to last 10 years, estimated to last over 100 hours with our system power draw. Each part is interchangeable - the lithium battery can be replaced with a 23lb lead acid battery, saving $200. The LimeSDR Mini can be eschewed for no SMS capability, saving $180. We will try and get SMS broadcast working with a second RTL-SDR, but so far we have only be successful in sniffing gsm, not broadcasting.

Social Impact

> *How well-tailored is the solution to the needs of the community and users for which it is designed? How will the design of the solution help engage community members in order to maximize utilization?*

The system is designed to be useful year round, and only *extra* useful during emergencies. By not perscribing a single way to use it, allowing for a channel-style message board allows the community to use things as they see fit. We will be working with Mozilla and the NSF to identify communities that have low connectivity, are at risk or have a history of infrastructure going down (i.e. flood zones, underdeployed power distribution), and working with local community leaders to show how it can help to own their own infrastructure.

In addition, documentation for all software and hardware will be updated or created to provide for an onboarding for new programmers / technicians, inspired by the Red Hook Initiative's digital stewards program, allowing for people interested in building skills in their own communities to grow. There is no reliance on a single provider for any component (even the iridium network can be replaced with LoRa dead-reckoning, for example), and no reliance on external guidance for the project to continue as long as it is useful.

Scalability

> *How will the solution be adaptable to a broader set of communities or areas? How scalable is the solution? How will you provide tools and documentation to anyone who might wish to build upon your work or launch a similar effort?*

The biggest scalability limit is human. The GuiFi network has over 35,000 active nodes. In addition to providing a comprehensive guide to building a node, we will be making sure each project has up-to-date onboarding documentation on how to contribute. This was part of our guidance of going for welcoming communities like the Raspberry Pi, over perhaps more powerful single board computers, for example.

Everything is available off-the-shelf, at the very least via amazon, without the need for special partnerships.

Openness

> *Mozilla works in the open. How will you document and share your project progress with the community? What documentation and resources have you created to help others understand and leverage your design in their own work?*

We publish in open formats, and have a strong culture of contributing back upstream, not only to the original software, but writing our own packages and getting them upstreamed into mainstream repositories like raspbian and homebrew, for example. We will share every WIP step publicly, through updating code repositories, writing blog posts, and livestreaming our work. In the past, we have had a great time participating in both geekstreams when building hardware projects, and twitch.tv/creative while programming emulators. The communities have been crucial in helping get past roadblocks, etc.

Portability

> *How portable is your proposed solution? Could your solution be easily carried by hand? Please explain.*

Its meant to be carried, or wheeled by a single person. The total system weight is about 12 lbs, 18lbs with solar panel. Unfortunately this is kind of expensive - when working with different communities, we will see if the tradeoff of adding 18lbs is worth the ~$200 saved.

Power

> *Could your solution be powered, for hours or days, by a portable power source? Please explain.*

The system includes a battery that should last 100 hours on a full charge. The solar panel should be enough, with 6 hours of direct sunlight, to last 18 hours.

Applications

> *What information, apps, services, software, etc. will your solution provide access to? Are these designed in a way that maximizes usability for the intended users?*

With an upstream internet connection, the system works fully with the regular internet. We may provide utilities such as DNS level adblocking/user-tracking filtering (using PiHole) to help reduce bandwidth and cpu load.

The forum software will work as normal. The SMS gateway will be default OFF until we notice that local upstream providers are not working, and will only turn on as a backup.

It is crucial that users do not have any additional cognitive load during an emergency to be able to communicate. Where possible, we should be using the same UX they are used to.

Previous Awards

No previous awards have been given.