pika has a locker (storage space in OpenAFS provided by MIT IS&T) named "zelda" that has some stuff in it. It's mostly pre2015, but it is not totally useless.

You can read more about OpenAFS and lockers at SIPB's AFS and You page.

Note: this page is from 2015(?) or something like that, it may or may not be at all relevant.

Sun Rack cps1500avr 1500
1: LCD
2: 2U Sunfire PSU 1 of 2
3: Netgear
4: 1U Sunfire PSA
Note: Not 100% sure on these.
Have email in to get second USB to Serial converter. Then set up get and bogomips. Make sure loongson2f-1 backups are up to date.

Cisco Rack APC 750
1: Netgear 1 of 3
2: Netgear 2 of 3
3: Netgear 3 of 3
4: Cisco Wireless
5: Cisco Router
6: V-P Fiber to Ethernet

We pay `rsync.net` $9/year to have 25G of backup space accessible over ssh and such.

We use public key authentication, and try to keep the authrorized keys list in sync with the list of yfncc keys in the yfncc-cdist repository. There is a password in the password stash in case things go wrong.

For `.ssh/config`:

```
Host zurich
    Hostname ch-s010.rsync.net
        User 16502
```

Currently, the backups contain the four yfncc git repositories; we should probably copy other stuff there as well.

Most pika computers are managed using `cdist`, using configuration files stored in git. `cdist` is a means of configuring the pika computers, particularly pika-web, as we choose. The current pika version can be found on the backups and is called `yfncc-cdist`

Relevant parts of cdist

The vast majority of times that yfncc needs to change something in `cdist`, it will be in `yfncc-cdist/cdist/conf/manifest`. Some of the most important items in this directory are highlighted here - `pika-web.mit.edu`: this script controls `pika-web`, `pika-wiki`, and `mealplan`. It contains all of the network settings (IP hostnames, MAC addresses, nameservers, ports, etc.) as well as numerous calls to other scripts an binaries to handle such these as certificate authentication. If you need to reinstall OpenBSD on pika-web, most of the initial configuration parameters can be found here.
- `bin/openbsd/`: This contains the various `go` binaries required to run the pika websites. They can be compiled from their corresponding repository on https://github.com/pikans. To update them, just make the changes in the appropriate repository, then compile them, and replace the binary in this directory. The `mealplan` repository has the most detailed documentation, and similar methods can be used on the other repositories.
- `pubkeys/`: This directory contains various public keys to give individuals access to the appropriate website-related user profiles. Of primary relevance here are `pika-private` and `yfnccs`. Any public key placed in the former will allow the user to use `ssh web@pika-web.mit.edu`. Any public key placed in the latter will allow the user to use `ssh root@pika-web.mit.edu`.

Updating cdist

1. Pull the current cdist state from the remote repositories. See the git page for details.
2. Make the desired changes on your local repository.
3. Commit your changes.
4. Push your changes
5. Navigate up to `yfncc-cdist/` in the cdist repo and run `./bin/cdist config -v pika-web.mit.edu`. Note that this requires that you have SSH keys for root on `pika-web.mit.edu`. If you do not have these, inquire with the current yfncc.
6. Restart the appropriate service(s). If you made significant changes to multiple services, it may be easier to restart the `pika-web` computer altogether, but beware that if you misconfigured things, `pika-web` may be unable to reconnect to the internet and will require you to manually fix it using the terminal in the pika basement.

MIT students should use https://certassist.mit.edu to get certificates.

When pikans are unable to get MIT client certificates (due to technical difficulties or not being current MIT affiliates), but still need access to the mealplan website or the wiki, we can give them client certificates signed by a pika certificate authority. The procedure is as follows:

First, make sure that the person's email address (lilah@example.com) is on pika-food or pika-private as appropriate.

Then:

```
$ ssh root@pika-web.mit.edu
# cd certs
# openssl req -new -newkey rsa:4096 -nodes -keyout lilah-key.pem -out lilah.csr
Country Name (2 letter code) []:US
State or Province Name (full name) []:Massachusetts
Locality Name (eg, city) []:Cambridge
Organization Name (eg, company) []:pika
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:Lilah Cat
Email Address []:lilah@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
# openssl x509 -req -days 3650 -in lilah.csr -CA pika-ca.pem -CAkey pika-ca-key.pem -CAcreateserial -out lilah.pem
# openssl pkcs12 -export -clcerts -in lilah.pem -inkey lilah-key.pem -out lilah.p12
# exit
$ scp pika-web.mit.edu:certs/lilah.p12 .
```

Now give lilah.p12 to the pikan, and help them import it into their web browser (for Firefox you go into Preferences -> Advanced -> Certificates -> View -> Import). During the last `pkcs12 -export` command you have the option to enter a password; this is probably overkill, but if you do set a password, you'll need to give it to the pikan when they import the certificate into their browser. (If you leave the password blank you might need to enter a blank password when importing.)

Installation instructions from certassist:

~~~~~~~ instructions-chrome/IE-windows: Open the downloaded *.p12 file, and follow the steps of the Certificate Import Wizard. instructions-chrome-linux: Save the* .p12 file, then import it at Settings â†’ Advanced â†’ Manage certificates â†’ Your Certificates â†’ Import. instructions-firefox-windows: Save the *.p12 file, then import it at Options â†’ Advanced â†’ Certificates â†’ View Certificates â†’ Your Certificates â†’ Import. instructions-firefox-mac/linux: Save the* .p12 file, then import it at Preferences â†’ Advanced â†’ Certificates â†’ View Certificates â†’ Your Certificates â†’ Import. instructions-chrome-android: Open the downloaded *.p12 file, and follow the prompts to install your certificate. instructions-chrome-mac: Open the downloaded* .p12 file, and follow the prompts to add your certificate to the login keychain. ~~~~~~~

How to set this up from scratch

If data from pika-web has been lost and you need to recreate this setup, here's how to create a new self-signed pika CA:

```
$ ssh root@pika-web.mit.edu
# mkdir certs
# cd certs
# openssl req -x509 -newkey rsa:4096 -nodes -keyout pika-ca-key.pem -out pika-ca.pem -days 3650
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:Massachusetts
Locality Name (eg, city) [Default City]:Cambridge
Organization Name (eg, company) [Default Company Ltd]:pika
Organizational Unit Name (eg, section) []:yfncc
Common Name (eg, your name or your server's hostname) []:pikans.org
Email Address []:yfncc@mit.edu
# exit
$ cd yfncc-cdist
$ scp root@pika-web.mit.edu:certs/pika-ca.pem cdist/conf/manifest/pubkeys/pika-ca.pem
$ git commit cdist/conf/manifest/pubkeys/pika-ca.pem -m "replace pika CA"
$ git push
```
~~~~~~~

Most pika computers are managed using `cdist`, using configuration files stored in git. All machines are meant to be accessed over `ssh` as root with public key authentication, there is an emergency password stash inside the leftmost server rack.

`pika-web.mit.edu` hosts `pikans.org`, `wiki.pikans.org`, and `mealplan.pikans.org`. The operating system is OpenBSD, the web frontends use some custom `go` code to glue together `golang` standard library implementations of `TLS` and `HTTP` with `rsc.io/letsencrypt` certificate manager (see https://github.com/pikans/simple-https-server/blob/master/main.go#L1). Access to the wiki is controlled using the `pika-wiki` moira group and client certificates from either MIT or StartCom; this is implemented as a HTTP reverse proxy (https://github.com/pikans/listproxy/blob/master/main.go#L1). The wiki itself runs Gollum, the software behind github wikis. Mealplan uses a similar frontend architecture, but the webapp itself is also written in go (https://github.com/pikans/mealplan). We should probably deduplicate and package the web frontend code some day.

`bogomips.mit.edu` is a storage server. It is used for archived pika stuff, backups, and various media. The operating system is FreeBSD, chosen due to the reliability features of `ZFS`.

`pika-guest-manager.mit.edu` does not exist, it never gives dhcp addresses (registered to yfncc) to guests plugged into some select ethernet ports or connected to `pika-` wifi. The operating system is OpenBSD resflash.

`pika-guest-ap-murph.mit.edu` is an openwrt wireless access point, currently disabled. It used to be connected to pika-guest-manager.mit.edu.

`musika.mit.edu` is a musicazooing server hooked up to the soundsystem in the Jazz Lounge. The operating system is Debian. When Musika breaks, fixing it usually works.

Currently, the machines that run Windows do not seem to be maintained by anybody. `get.mit.edu` is accessed over Windows RDP and runs treasuring software (Quickbooks Pro 2015). This is somewhat documented in Treasuring System Administration.

Public Repositories

Public repositories are at https://github.com/pikans. This includes the mealplan server, the web server, the wiki server, the public wiki server, and a helper script for SHITS reminders. You can clone all of them by doing

```
curl -s https://api.github.com/users/pikans/repos | grep \"clone_url\" | awk '{print $2}' | sed -e 's/"//g' -e 's/,//g' | xargs -n1 echo git clone
```

Private Repositories

All pika private computer stuff is in git repositories: the website, the wiki content, and the cdist configuration files for all pika-managed computers. All git repositories are synchronized to two locations: `bogomips.mit.edu:/left/archive/$REPO` and `16502@ch-s010.rsync.net:$REPO` as a backup. In addition to that, the active copies of the website and the wiki are served from `web@pika-web.mit.edu:/home/web/pikawiki` and `pika-web.mit.edu:/home/wiki/pikawiki` respectively.

When editing the content of a repository, one should always push the change to all locations (as done by default in any configuration on this page). Any one of the pushes failing should be reported to yfncc and we will fix it :). Also, you probably need permissions to access any of these repositories -- just ask. Pull from the one that makes sense.

pika-website

This repository is used to update the pika website (just push to the webserver); it also contains archived versions of the website. To use it, run `git clone ssh://web@pika-web.mit.edu:/home/web/pika-website` and then edit `pika-website/.git/config` to the following:

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "web"]
    url = web@pika-web.mit.edu:/home/web/pika-website
    fetch = +refs/heads/*:refs/remotes/web/*
[remote "bogomips"]
    url = bogomips.mit.edu:/left/archive/pika-website
    fetch = +refs/heads/*:refs/remotes/bogomips/*
[remote "zurich"]
    url = 16502@ch-s010.rsync.net:pika-website
    fetch = +refs/heads/*:refs/remotes/zurich/*
[remote "all"]
    url = web@pika-web.mit.edu:/home/web/pika-website
    url = bogomips.mit.edu:/left/archive/pika-website
    url = 16502@ch-s010.rsync.net:pika-website
    fetch = +refs/heads/*:refs/remotes/web/*
[branch "master"]
    remote = all
    merge = refs/heads/master
```

Running 'git push' with the above configuration produces this output, and prompts you to enter the root password for rsync.net user 16502.

```
your_user$ git push
Counting objects: 3, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To pika-web.mit.edu:/home/web/pika-website
    540d65b..486c515  master -> master
Counting objects: 3, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To bogomips.mit.edu:/left/archive/pika-website
    540d65b..486c515  master -> master
Password: [ENTER RSYNC PASSWORD HERE]
Counting objects: 3, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To ch-s010.rsync.net:pika-website
    540d65b..486c515  master -> master
```

To pull updates to an already downloaded repository, run `git pull --rebase web master`.

(not ideal route) If that doesn't work, run `git checkout HEAD -- [filename.ext]`

yfncc-cdist

This repository contains autoconfiguration files for all pika computers (in directory `conf/cdist/manifest`). To clone it, run `git clone ssh://root@bogomips.mit.edu/left/archive/yfncc-cdist` and then edit `yfncc-cdist/.git/config` to the following:

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true

[remote "bogomips"]
    url = bogomips.mit.edu:/left/archive/yfncc-cdist
    fetch = +refs/heads/*:refs/remotes/bogomips/*
[remote "zurich"]
    url = 16502@ch-s010.rsync.net:yfncc-cdist
    fetch = +refs/heads/*:refs/remotes/zurich/*
[remote "all"]
    url = bogomips.mit.edu:/left/archive/yfncc-cdist
    url = 16502@ch-s010.rsync.net:yfncc-cdist
[branch "pika"]
    remote = all
    merge = refs/heads/pika

# "midstream" for sending patches to cdist itself

[remote "github"]
    url = git@github.com:andres-erbsen/cdist
    fetch = +refs/heads/*:refs/remotes/github/*
    merge = refs/heads/pika
[branch "midstream"]
    remote = github

# upstream, cdist configuration management

[remote "origin"]
    url = https://github.com/telmich/cdist.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
    remote = origin
    merge = refs/heads/master
```

To pull updates to an already downloaded repository, run `git pull --rebase bogomips`.

pikawiki

This repository contains the content (not the implementation) of the wiki. You can commit and push to the wikiserver, and the wiki content will change. To clone it, run `git clone ssh://wiki@pika-web.mit.edu:/home/wiki/pikawiki` and then edit `pikawiki/.git/config` to the following:

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "wiki"]
    url = wiki@pika-web.mit.edu:/home/wiki/pikawiki
    fetch = +refs/heads/*:refs/remotes/wiki/*
[remote "bogomips"]
    url = bogomips.mit.edu:/left/archive/pikawiki
    fetch = +refs/heads/*:refs/remotes/bogomips/*
[remote "zurich"]
    url = 16502@ch-s010.rsync.net:pikawiki
    fetch = +refs/heads/*:refs/remotes/zurich/*
[remote "all"]
    url = wiki@pika-web.mit.edu:/home/wiki/pikawiki
    url = bogomips.mit.edu:/left/archive/pikawiki
    url = 16502@ch-s010.rsync.net:pikawiki
[branch "master"]
    remote = all
    merge = refs/heads/master
```

To pull updates to an already downloaded repository, run `git pull --rebase wiki master`.

facebook_archive_2021_04

This repository contains the archive of pika's facebook account, as of April 2021, including all images and posts. To clone it, run `16502@ch-s010.rsync.net:facebook_archive_2021_04` and then edit `pikawiki/.git/config` to the following:

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "zurich"]
    url = 16502@ch-s010.rsync.net:facebook_archive_2021_04
    fetch = +refs/heads/*:refs/remotes/zurich/*
[remote "all"]
    url = 16502@ch-s010.rsync.net:facebook_archive_2021_04
[branch "master"]
    remote = all
    merge = refs/heads/master
```

To pull updates to an already downloaded repository, run `git pull --rebase zurich master`.

To request a public IP/hostname from IS&T (for something you want to be accessible outside MIT), you can file an IS&T helpdesk ticket with your desired hostname and your device's MAC address, or you can repurpose one of the hostnames here (you'll need to change your device's MAC) and get yfncc to change the name to whatever you want.

spare hostnames available for use:

- ziyou.mit.edu, 18.18.214.18, MAC 00:23:8b:35:6e:74

pika has a lot of mailing lists for various purposes. It's useful to have a list of these to reference when you want to send emails, so here ya go. It seems like a lot, but mostly you should care about the main/social lists and reference the elected positions when you need:

(all are @mit.edu; for example, pika-private is pika-private@mit.edu.) (Also, all of these except "pikans" are moira managed lists; you can add/remove yourself at https://groups.mit.edu/webmoira/. Requires certificates)

Email List Etiquette

- Be careful what you send to **pika-social** - it has a whole lot of people on it, many of whom we may want to rush at some point, so we want to make a positive impression.
- To that effect: emails like "who left their laundry in the washer" should ALWAYS go to **pika-private** and should NEVER go to **pika-social**

Getting added to lists

Please in this order:

1. Try to add yourself - click on the link next to the list name.
2. Bug yfncc@mit.edu to add you

Administering Lists

- When making a new moira list for pika, make sure the list ownership includes at least pika-private@mit.edu and yfncc@mit.edu. You can set the owner to pika-private-plus-yfncc@mit.edu if that's all you need, or create another list that owns your new list and includes pika-private and yfncc along with any other owners it should have.
- Moira lists - the links below will let you admin lists, or use "blanche" from the command line.
- Mailman lists - click on the link below, then scroll to the bottom and click on the " administrative interface (requires authorization)" link. You'll need the admin password, some of which are in `/private/mailinglist-passwords`.
- Note: starting March 2013, records of past pika-private membership gets dumped in /mit/zelda/pika_private_snapshot.log on AFS, in case you are curious who lived here when. [Note: as of 2019, this link may be defunct. Oops.]

List of lists

Main/Social Lists (all residents are on all of these):
pika-private: Residential in-house mailing list for logistical/house-only-relevant matters. Medium/high traffic
pika-commie: (Commie as in "community") Social/random discussion list for close friends of the house (arbitrarily defined). Medium/high traffic
pika-social: Social list for large events and meal-plan signups. Low traffic.
pika-food: Dinner and mealplan related announcements and discussion. Request kitchen duty swaps and announce late dinners here. Medium traffic

Elected Positions (yfn stands for 'your friendly neighborhood'):
yfnhm: House managers; email us if stuff in the house is broken or if you're gonna miss SHITS or something.
pika-treasurer: (alternatively yfnt) They handle the money. Talk to them if you need to set up a housebill payment plan.
yfnkm: Kitchen Manager, handles mealplan scheduling and signup periods + general kitchen stuff.
yfns: Stewards, they get the food.
pika-rush: Rush/recruitment; they organize rush week and bid meetings.
pika-speaker: MIT makes us have a "president" to send to meetings; we call this position "speaker". Not really any reason you should email this list.

Alum Lists (we should be better about adding alums to these!):
pikans: Theoretically a list of everyone who has lived at pika ever. Very low traffic (2-3 emails/year)
pika-alum-talk: Alum general-discussion list; people are added on request. Still pretty low traffic.
nutso-flaming-pikans: Theoretically more spammy version of pika-alum-talk. Medium/low traffic.
pika-local-alums: Theoretically pika alums who live in the Cambridge/Boston metro area.

Other stuff:
pika-lynx: People sometimes send random web links here (funny youtube videos! insightful speeches! etc.)

List of Lists [as it was in 2013. As of 2019, most of these are no longer extant.]

- Please add any lists you know of that are not listed!
- Feel free to update below with more current / correct information :-)

| List (admin link) | Description | Note | Elected? | Active? | |---|--|--|----------|---------| | bay-area-pikans 1 | pikans living in the bay area of California | | | | | desi 2 | meow! | | | | | fsilg-sysadmin 3 | FSILG System Administration Collaboration | | | | | fsilg-sysadmin-archives 4 | FSILG System Administration Collaboration -

Archives | | | | | housecorp 5 | Alumni Corporation of pika | | | | | in-house-pikans 6 | forwards to pika-house | | | | | jazz-lounge 7 | | | | | | joepika 8 | | used for common login to various web services | | | | kingpin 9 | The pika kitchen kingpin! | | | | | mealplanner-dev 10 | hippies working on pika's lates server | | | | | mountain-pika 11 | pika pika pika pika climbing | | | | | nutso-flaming-pikans 12 | Random commentary for pikans and their friends. | | | | | ny-pikans 13 | pikans living in the NYC area | | | | | out-house-pikans 14 | forwards to pika-alum-talk | | | | | pika-alum-talk 15 | Present and former members of pika and their friends; higher traffic than pikans@mit.edu. | | | | | pika-ambassador 16 | | single point of contact for pika's neighbors | | | | pika-barn 17 | pika barn renovations coordination | | | | | pika-bids 18 | People with bids at pika who aren't frosh | | | | | pika-bids-10 19 | | | | | | pika-books 20 | | people working on jhamrick's pika books webapp | | | | pika-bread 21 | | | | | | pika-bridge 22 | | | | | | pika-car-czar 23 | | person who deals with parking / cars | | | | pika-chickens 24 | | | | | | pika_chickens 25 | | pika mailing lists use dash, not underscore; so it's preferable to use one of the other aliases of this list. | | | | pika-chicks 26 | | same list as pika-chickens | | | | pika-chicks-request 27 | | | | | | pika-climbers 28 | climbers | | | | | pika-commie 29 | people who are around pika a lot | list for **current** pika community | | | | pika-construction 30 | Discussion of Construction at pika | | | | | pika-cooks 31 | pika big cooks | | | | | pika-csa 32 | | people who get notified by the scary lady | | | | pika-cycling 33 | | | | | | pika-daf 34 | Development of pika's undergraduAte rush effort, Fuck yeah | | | | | pika-dialup-acl 35 | ACL for dialup users of pika cluster machines | | | | | pika-disc 36 | pika ultimate frisbee! | | | | | pika-errors 37 | The list of people we really shouldn't have given a bid to. | | | | | pika-food 38 | pika meal plan folk | | | | | pika-foot 39 | soccer with pikans | ... or pika foot fetishists? | | | | pika-frosh 40 | Frosh with bids at pika (non-frosh are on pika-bids) | | | | | pika-fucking-social 41 | | | | | | pika-gardens 42 | | pika gardens and general yard maitenance/projects | | | | pika-gatekeeper 43 | | | | | | pika-graywater 44 | | | | | | pika-green 45 | | | | | | pika-groupies 46 | pika "groupies" | | | | | pika-historian 47 | pika historian | | | | | pika-house 48 | non-private pika-related discussion (includes friends of the house) | | | | | pika-house-managers 49 | | current and interested former house managers | | | | pika-house-managers-current 50 | current pika house managers - alias for yfnhm lisr | | | | | pika-interest 51 | If you have visited pika during a rush period... | | | | | pika-irdf 52 | Fully utilize IRDF Grant money for pika | | | | | pika-jam 53 | | | | | | pika-jazz 54 | A list about live shows, centered around jazz but including other genres as well | | | | | pika-list-of-lists 55 | | | | | | pika-local-alums-announce is DEFUNCT / DEPOPULATED as of 5/1/2013 56 | announce things to pika alums who live close to pika | rexaisecec | | | | pika-lynx 57 | Opt-in pictures of cats and occasional discussion... | | | | | pika-math 58 | in which pika learns math | | | | | pika-meditation 59 | | | | | | pika-mp3-announce 60 | folks interested in pika's internal mp3 server | | | | | pika-music 61 | rockin music lovin pikans | | | | | pika-neighbors 62 | | After people sign up it's useful to login there to put in their full names. | | | | pika-odd-jobs 63 | | | | | | pika-orgies 64 | | | | | | pika-orgy 65 | | | | | | pika-overlord 66 | | | | | | pika-pledges 67 | people moving into pika soon | | | | | pika-power 68 | | | | | | pika-president 69 | Pika President | | | | | pika-private 70 | pika housemember-only discussion | | | | | pika-privates 71 | | | | | | pika-produce 72 | For ordering produce | | | | | pika-questions 73 | pika, an independent living group of MIT | For people who want to answer questions about pika | | | | pika-rush 74 | | | | | | pika-rush-2010 75 | freshman rushees fall 2010 | | | | | pika-secretary 76 | | | | | | pika-social 77 | pika's social list | | | | | pika-social-request 78 | | | | | | pika-spang 79 | | | | | | pika-summer 80 | summer pikans; use instead of summer-pikans | | | | | pika-summer-dictator 81 | | | | | | pika-summer-food 82 | pika summer meal plan | | | | | pika-summer-high-muckymucks 83 | pika summer officers | | | | | pika-summer-interest 84 | People interested in living at pika, summer 2009 | | | | | pika-tkd 85 | | | | | | pika-treasurer 86 | | | | | | pika-vote 87 | | | | | | pika-vps 88 | | | | | | pika-vps-root | | | | | | pika-windows 89 | | | | | | pika-zipcar 90 | | | | | | pikachuoftheday-request 91 | | | | | | pikans 92 | Present and former members of pika and their friends. (Low traffic!) | | | | | pikans-acl 93 | pikans ACL | | | | | pikans.org-mesh 94 | | Notifications and discussion for pikans.org/mesh-* network | | | | rihp 95 | forwards to pika-private | | | | | rihp-plus-pledges 96 | pika rihp and pledges | | | | | seattle-pikans 97 | pikans living in the Seattle area | note | | | | summer-dictator 98 | pika summer dictator | | | | | summer-pikans 99 | Undepricated to avoid confusion errors; everyone should be using spam-filters these days | | | | | uk-pikans 100 | pikans living in the UK | | | | | yfnac 101 | | your friendly neighborhood alum chair | | | | yfncc 102 | | your friendly neighborhood computer chair | | | | yfncc-and-rush 103 | | | | | | yfncc-logs 104 | yfncc-related logs | logs from computers managed by yfncc | | | | yfnhm 105 | Your friendly neighborhood House Managers | | | | | yfnj 106 | Your Friendly Neighborhood Jo(h)n | | | | | yfnkm 107 | your friendly neighborhood kitchen manager at pika | | | | | yfns 108 | Your Friendly Neighborhood Stewards | | | | | zelda 109 | was user group, now belongs to locker | acl for the zelda openafs locker | | |

How to try fix temporary fuckage by updating software:

1. get a former yfncc (as of 2019: kruerj@reed.edu, joebob, dukhovni, and andreser are all options. Ask those listed earlier first :) to add your SSH key to musika

2. go to the seekrit yfncc password list and find the root password for musika

3. ssh into musika

4. go into the "musicazooing" folder and run "install.sh", entering the password when prompted:

```
$ cd musicazooing $ ./install.sh
```

This will upgrade all the system packages and all the python packages that musicazoo depends on. In particular, it's important to keep the youtube downloader package up to date, because google keeps changing the way youtube works and then the old version of the downloader won't work anymore.

1. Reboot musika

2. Jam out to sick beats!

pika has one Ubiquity NanoBridge on its roof.

- NanoBridge M2 https://m2.pikans.org (2.4 GHz) WLAN0 MAC 00:27:22:30:6A:7B - LAN0 MAC 00:27:22:31:6A:7B

Old note `There is a 2nd M2 that should be on PBE's roof soon.`

Doc

Documentation is available.

Notes

```
p25

Router or SOHO Router mode keeps broadcast traffic
within its respective broadcast domain, so that
broadcast traffic will not overload the overall
traffic in the network.

Bridge The device acts as a transparent bridge and
operates in Layer 2, like an unmanaged switch.
There is only one IP address for the device in Bridge
mode.

p26

Auto IP Aliasing

If enabled, automatically generates an
IP address for the corresponding WLAN/LAN interface.
The generated IP address is a unique Class B IP address
from the 169.254.X.Y range (netmask 255.255.0.0), which
is intended for use within the same network segment only.
The Auto IP always starts with 169.254.X.Y, with X and Y
as the last two octets from the MAC address of the device.
For example, if the MAC is 00:15:6D:A3:04:FB, then the
generated unique Auto IP will be 169.254.4.251.

The Auto IP Aliasing setting can be useful because you
can still access and manage devices even if you lose,
misconfigure, or forget their IP addresses. Because an
Auto IP address is based on the last two octets of the MAC
address, you can determine the IP address of a device if
you know its MAC address.

p28

Bridge Network

(Available in Advanced view.) You can create one or
more bridge networks if you need complete Layer 2
transparency. This is similar to using a switch â€“ all traffic
flows through a bridge, in one port and out another
port, regardless of VLANs or IP addresses. For example,
if you want to use the same IP subnet on both sides of a
device, then you create a bridge network. There are many
different scenarios that could require bridged interfaces,
so the Bridge Network section is designed to allow
flexibility.

Click the + button to display the Bridge Network section

p30

Router

In Router mode, the device operates in Layer 3 to perform
routing and enable network segmentation â€“ wireless
clients are on a different IP subnet. Router mode blocks
broadcasts and can pass through multicast packet traffic.

The device can act as a DHCP server and use Network
Address Translation (Masquerading), which is widely used
by APs. NAT acts as the firewall between the LAN and
WAN.

In Router mode, the WLAN functions as the Wide Area
Network (WAN). The Ethernet ports function as the LAN.
Each wireless or wired interface on the WAN or LAN has
an IP address. For example, the following diagram shows
the NanoStation at a residence wirelessly connecting to a
WISP tower.

p48

Sensitivity Threshold, dBm Defines the minimum client
signal level accepted by the AP for the client to connect.
If the client signal level subsequently drops, the client
remains connected to the AP.

p49

Signal LED Thresholds
```

(This feature is not available on all devices.) You can configure the LEDs on the device to light up when received signal levels reach the values defined in the following fields. This allows a technician to easily deploy an airOS CPE without logging into the device (for example, for antenna alignment operation).

p58

Site Survey

The Site Survey tool searches for wireless networks in range on all supported channels. In Station mode, you can change frequency list.

The nanobridges support arpnat, but don't have a great description of what it is; here is a better description.

What is ARPNAT?

ARPNAT is a clever hack. Itâ€™s the same idea as NAT for IP networks, except it works one layer deeper. Instead of translating IP network addresses, the router translates between the MAC hardware addresses on each side of it.

If something on the wired side of the router makes an ARP request for the MAC address of an IP on the wireless side, then the router forwards the request as if it came from the router. When the response comes back, it mangles that too. Instead of passing back the real MAC (which lives on the wireless network), the router gives its own wired MAC address. Then, when it receives frames for IP addresses on the wireless network, it forwards them through. It does this to both sides of the bridge.

Hereâ€™s an example, taken from my network:

1. XBMC Xbox asks â€œWhat is the MAC address of 10.0.0.1? Tell me: 10.0.0.5.â€ This is a broadcast ARP request over the wired ethernet.
2. Asus router sees the ARP request, and forwards the broadcast onto the wireless network, as â€œWhat is the MAC address of 10.0.0.1? Tell me: 10.0.0.4.â€
3. Access point (at 10.0.0.1) gets the ARP request from the wireless, and replies with its MAC address: AA:AA:AA:AA:AA:AA.
4. Asus router gets the ARP response, sent to 10.0.0.4. It mangles it and forwards it back to the Xbox as â€œ10.0.0.1 is at BB:BB:BB:BB:BB:BBâ€, where BB:BB:BB:BB:BB:BB is the wired address of the Asus itself.
5. Now, any time the Xbox wants to talk to 10.0.0.1, it will talk to BB:BB:BB:BB:BB:BB, and the Asus will forward that frame onto the wireless network, to the real destination AA:AA:AA:AA:AA:AA.

Configuration instructions for Netgear GS748 switches

These are `pika-no-such-sw-3.mit.edu`, `pika-no-such-sw-4.mit.edu`, and `pika-no-such-sw-5.mit.edu`; see the list of machines

1. Use a paperclip to hold down Factory Defaults reset button (on right side) for five seconds. Wait for the reboot.
2. Connect to it on one of the ethernet ports using a laptop. Set a static IP address in the `192.168.0.*` subnet, with subnet mask `255.255.255.0`. Default gateway shouldn't really matter but you can put the switch's IP:`192.168.0.239`.
3. Open a supported browser (Firefox version <=3.8; Chrome seems to work now) to `192.168.0.239`. Log in with the default password "password".
4. Fill in system information.
5. Set the time.
6. Set a secure password.
7. **IMPORTANT** for not getting IS&T yelling at you: Disable Spanning Tree Protocol. Switching > STP > Global Settings > Spanning Tree State -> Disabled; BPDU Handling -> Filtering.
8. Register it using command `moira` in Athena: machines > add; network "pika"; assign it the desired IP (or [unique] if you want it to pick)
9. Upgrade firmware (if necessary):
10. Download from Netgear
11. Unzip
12. Maintenance > Download > Via HTTP > Choose the `.ros` file
13. Apply
14. Reboot (reset) switch
15. Set IP configuration:
16. Static IP address
17. Use the IP address assigned above
18. Subnet mask `255.255.255.0`
19. Gateway: whatever IP `pika-rtr-ether.mit.edu` points to

David Lawrence <dlaw@dlaw.me>

10/20/2016 6:17 AM

Re: [Ailg-it] Fwd: [mitailg] FSILG network switch upgrade announcement

To Tim Gorton <tim@timgorton.com> Copy ailg-it <ailg-it@mit.edu>

Dear Tim,

Your house is welcome to attach any devices it likes to your Ethernet drops. What MIT is saying is that you must purchase at least one standard-issue new switch in order for those Ethernet drops to remain available. The IT committee does not recommend that you continue to rely on older switches as they are likely to fail soon. But nothing will prevent you from attaching them to the new switch if you would like.

For completeness, I should note that MIT IS&T does have a policy on the books which forbids the use of external switches. My understanding is that this policy exists primarily to prevent students in dormitories from creating routing loops and bringing the network down for an entire building. It is not presently enforced except in case of a network issue and it should not be an issue for you.

We do recognize the burden imposed by the purchasing requirement, which is why [illegible] arranged [illegible] better than 50% subsidy, free installation, and included 5 year warranty. It comes out to [illegible] $5 per resident per month which is not bad for the level of wired service provided.

Best,

David Lawrence
IT committee chair

On Oct 20, 2016 8:54 AM, "Tim Gorton" <tim@timgorton.com> wrote:

> A question here - is MIT saying they'll cut us off unless we replace every piece of (perfectly well-functioning) network hardware in the building with their new switches, or that we need at least a single new one in the building, even if older switches are still connected to it?
>
> thanks,
>
> TG
>
> ADP Corp Treasurer
>
> ---------- Forwarded message ----------
> From: David Lawrence <dlaw@dlaw.me>
> Date: Thu, Sep 22, 2016 at 10:00 PM
> Subject: [mitailg] FSILG network switch upgrade announcement
> To: <mitailg@mit.edu>
> Cc: ailg-it <ailg-it@mit.edu>
>
> Dear FSILG members and AILG stakeholders,
>
> For the last several years, the AILG IT committee has been working with MIT's IS&T department to upgrade FSILG network infrastructure to the state-of-the-art and to integrate it with the on-campus network. In 2014, we completed a major milestone with the installation of direct fiber-optic connections from FSILGs back to MIT. Latest-generation wireless access points were deployed in 2015. And earlier [illegible] we made MIT telephone numbers and handsets available to [illegible] FSILG.
>
> Now we are pleased to announce an upgrade program for the wired network switches in every FSILG. The current Netgear switches are the last remaining hardware from the original [illegible] and are nearly [illegible] years old. They have never been officially supported by MIT IS&T, and as their [illegible] houses [illegible] wired networks offline with no warning and no responsible party to provide [illegible] are already [illegible]

Current state

Machines

Some of the switches we have:

| Hostname | IP | Model | Description | | -------- | ---| ----- | ----------- |`pika-003t-sw-1.mit.edu` | `10.102.214.3` | Cisco C3KX-NM-10G | For IS&T-managed wifi routers (not for our use) | | `pika-003t-sw-2.mit.edu` | `10.102.214.2` | Cisco Catalyst 3650 | New switch from Dec 2016 "upgrade". Managed by IS&T; we can plug things into it | | `pika-no-such-sw-3.mit.edu` | `18.102.214.10` | Netgear GS748TP | Unofficial pika switch; on top of networking rack; **not supported by IS&T**; used to be pika-cnup-sw-1.mit.edu when IS&T installed it | | `pika-no-such-sw-4.mit.edu` | `18.102.214.11` | Netgear GS748TP | Unofficial pika switch; on top of networking rack; **not supported by IS&T**; used to be pika-cnup-sw-2.mit.edu when IS&T installed it | |`pika-no-such-sw-5.mit.edu` | `18.102.214.12` | Netgear GS748TP | Unofficial pika switch; on top of Sun rack; **not supported by IS&T**; used to be pika-cnup-sw-3.mit.edu when IS&T installed it |

[[yfncc/network/TopOfNetworkRack.jpg]]

mitnet

PIKA connects to internet through MIT, managed by IS&T. The subnet `18.102.214.0/24` is routed to the building. Addresses from `18.102.214.10` to `18.102.214.100` are statically assignable (moira network `PIKA`, owned by `ht-pika-acl`). The MIT-wide captive portal and DHCP system seems to be assigning IP-s from `18.102.214.101` to `18.102.214.250` (moira network `PIKA-DYN`, owned by `network-acl`).

The house is connected to the MIT network using a Lightower fiber connection that enters through the wall next to the window by the racks. The fiber box is in the middle of the network rack and the computer rack next to the studyroom door. An optical goes from the fiber box to a IS&T router (Cisco Catalyst 4948E). One port on the IS&T router is reserved for the IS&T wireless controller right above the router (`pika-003t-sw-1.mit.edu`). The good access points are owned and managed by IS&T.

After the IS&T router, the new IS&T switch (`pika-003t-sw-2.mit.edu`) is used in combination with two of our own switches (`pika-no-such-sw-3.mit.edu`, `pika-no-such-sw-4.mit.edu`) to distribute ethernet to the house. **These two switches are not supported by IS&T**. The hardware is Netgear gs748tp, and it has usual "smart" equipment issues. For example, only old browsers are supported on the web-based admin interface; Firefox portable 3.5.6 seems to work. Chrome 55 (up-to-date as of Dec 2016) seems to work fine too.

The switches are connected to a breakout board on the high half of the network rack, whose ports map to physical locations in the house (both ends of the same wire are labeled with the same combination, e.g. `54d1`). Some breakout ports are wired through a power-over-ethernet injector for pika-owned access points.

IS\&T KB: FSILG network home page has more (generic) information.

The pikan registered as "FSILG network contact" at IS&T can register new IP addresses to the pika subnet. It is not possible to give this power to the `yfncc` list.

History

Dec 2016: IS&T "upgrade"

IS&T required us to purchase a Cisco Catalyst 3650 and stopped supporting the CNUP switches. We were allowed to keep the switches but they are not supported by IS&T; we should be careful not to misconfigure them in a way that causes trouble for IS&T, and if something doesn't work, we can't ask IS&T for help.

Clarifying email from dlaw:

> Dear Tim,
>
> Your house is welcome to attach any devices it likes to your Ethernet drops. What MIT is saying is that you must purchase at least one standard-issue new switch in order for those Ethernet drops to remain available. The IT committee does not recommend that you continue to rely on older switches as they are likely to fail soon. But nothing will prevent you from attaching them to the new switch if you would like.
>
> For completeness, I should note that MIT IS&T does have a policy on the books which forbids the use of external switches. My understanding is that this policy exists primarily to prevent students in dormitories from creating routing loops and bringing the network down for an entire building. It is not presently enforced except in case of a network issue and it should not be an issue for you.
>
> We do recognize the burden imposed by the purchasing requirement, which is why we have arranged the greater than 50% subsidy, free installation, and included 5 year warranty. It comes out to about $3 per resident per month which is not bad for the level of wired service provided.

Best,

David Lawrence IT committee chair

2015: networking upgrade

We got rid of our Comcast connection and transitioned to only using IS&T-supplied fiber. IS&T upgraded it to gigabit speed. pika was told that the hardware could support up to 10 Gbps up/down bandwidth, but is currently software limited to 1 Gbps. We were told that if we could demonstrate a legitimate usecase for requiring more than the 1 Gbps allocated, we could probably get that limit raised. This has not appeared to be necessary as of December 2016.

2009: Community Network Upgrade Project

This gave us the three Netgear GS748TP switches which used to be `pika-cnup-sw-1.mit.edu`, `pika-cnup-sw-2.mit.edu`, and `pika-cnup-sw-3.mit.edu`. See the configuration instructions.

You can use the printer by going to your printer settings and adding a network printer with the address `stud.mit.edu` or the URI `socket://stud.mit.edu:9100`. It is a `HP LaserJet M607`. It should just work. You can only print to it if you are on MITnet (18.0.0.0/8, Stata, or MIT guest; VPN also works).

In case you are having configuration trouble, here is the section of `/etc/cups/printers.conf` that has been working for me for years (I made it for the previous printer, but it seems to work with the current one as well):

```
<Printer stud>
UUID urn:uuid:d58b1953-ca2c-3250-4ad4-07d9a86d5cc7
Info HP LaserJet 600 M601 M602 M603
Location stud.mit.edu
MakeModel HP LaserJet 600 M601 M602 M603 Postscript (recommended)
DeviceURI socket://stud.mit.edu:9100
State Idle
StateTime 1495721345
ConfigTime 1472592667
Type 8425684
Accepting Yes
Shared Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy stop-printer
</Printer>
```

yfncc onboarding

Y'argh so yer the new yfncc! Congrats matey!

There are two things you need to do right away: get your computer set up to administer pika's computers and read about pika's computing infrastructure.

Bootstrapping

1. Get your SSH keys on the pika computers. This will let you SSH (remotely log-in) into them later without having to remember any passwords. You should do this by generating an SSH key (run `ssh-keygen -t ed25519`) and sending your public key (usually `~/.ssh/id_ed25519.pub`) to a former yfncc (as of 2019: kruerj@reed.edu and joebob@mit.edu are the go-to people to ask.)
2. Clone all of the yfncc-relevant git repositories to your machine. This will let you do things like make changes to this wiki page and fix the mealplan software at the start of a new semester. Do this by following the instructions on the git page.
3. Make sure that you or another active yfncc are on ht-pika-acl@mit.edu; this is the list that makes you IS&T Network Contact and gives you bits to add hostnames/IPs to pika's network. A current member of ht-pika-acl can email ht-ilg-acl@mit.edu and request that you be added.
4. Maybe join the ailg-it@mit.edu list to hear about ILG IT stuff. http://mailman.mit.edu/mailman/listinfo/ailg-it

Required reading

- fixing musika
- pika computers
- pika network info

Read when necessary

- pika guestnet (broken link)
- certificates