Eric Darsow
Heinz College| Linux Independent Study Fall 2015 | Professor Dennis Moul

Linux Networking Project Table of Contents

Project Introduction (tl;dr;sentimental and reflective)

Nearly twenty years ago my father, a scrupulous electrical engineer working for then-proprietary-software-giant IBM, agreed to tenderly wrap the component of our family's 100-mhz Micron desktop computer in towls and drag them to a basement room at the local community college where the town's Linux User's Group (LUG) was having its monthly noob install help night. I had been counting down the days to that moment when the Linux guru helping us with the install booted the newly minuted Linux OS for the first time.

Now a *true nerd* would probably have elaborated on his or her worry about this or that bootloader configuration file that may not have been configured properly during the install. In reality, I was holding my breath in anticipation of the *relief* I so desperately craved from the worry that an install failure would mean my father would have to explain to my mother why she couldn't check her email or edit her online genealogy schematics.

Thankfully the install proceeded without catastrophe and our family's computer had not been rendered an expensive closet decoration. And, I should be fair: my delightful parents would certainly not have been all that upset if the Mandrake Linux install had fouled the system for a few days; we did have backup copies of all our important documents. After the stress subsided, I was living the nerd's dream: I had an open-source OS to play with at my leisure, in my very own home. As long as Windows 3.11 would still boot, I had the root password and could do anything I wanted. And, sadly, it has taken me about 18 years to realize what I never quite understood in middle school: the magic of Linux is in the *command line*.

So I dabbled with Linux here and there, compiling *hello world* variants in *C* and even toyed with learning all the key combinations of *emacs.* But in high school I was a decidedly a *chemistry junkie* and not a *computer geek*. Strangely, what felt like my true technical calling in life (which was at the time chemical engineering), gave way to international justice advocacy and four years of teaching English Language Arts to ninth graders. Linux took a back seat as I zoomed off to South Africa where I studied nonprofit organization development in a little town of East London for my undergraduate degree in social anthropology.

And yet, even when enmeshed in the politics of post-apartheid reunification policy, Linux was never fully out of sight. For example, I wanted to travel overseas without lugging around an expensive electronic device so I arranged with the tech master at the South African nonprofit to take home an old desktop running, of course, a dusty version of Linux. I used open source software to code my interview transcripts, which *felt good*, but I wasn't directing my energies at the exciting developments in the field.

As sacrilegious as this might sound, Linux also deserves due credit for being part of the landing pad which absorbed the shock from crash ensuing from my mad dash out the upper story window of the burning tower of K-12 education in the United States. I sent off applications to a few masters programs in information systems programs hosted in B-schools since I didn't have any actual tech or programming experience and landed here at CMU. In the past fifteen weeks I have reluctantly discovered that if I were to enter a classroom ever again, I would want to be teaching Linux math, or Java-- and decidedly *not* English. The stress and groaning incident to the hours I've spent learning about dozens of facets of operating systems and networking have spawned a profound sense of energetic excitement to continue to learn the language of unix-based operating systems and apply open source technologies to relevant social issues.

Project Introduction (technical goals and outcomes)

Robust learning is rooted in an ever-increasing familiarity with the entire spectrum of components that make up a system, such as an operating system or a computer network. Until this semester of serious study of Linux, I had not appreciated the degree to which a broad familiarity with with a range of tools can engage and inform more focused training in a given programming language (java, c, etc.) or tool set (GIS, machine learning). The goal of this independent study was to engage in meaningful ways with the range of technology used in computer networking. Hence, the following report will detail the process and outcomes associated with my building a PC from parts and configuring it into a functional LAMP to serve content to the Internet from a LAN routed and protected two by customized routers.

My main reference text is Mark Sobell's A Practical Guide to Ubuntu Linux. I read the first 600 pages on basic system administration and shell operations straight through and worked the examples. I also referenced his two other books, one specific to Red Hat/Fedora called A Practical Guide to Fedora and Red Hat Enterprise Linux 7th edition (2013), and one on general shell scripting, Perl, Awk, sed and a 400-page command reference called A Practical Guide to Linux Commands, Editors, and Shell Programming. *Sobell* is the guy!

I will begin with technical information about the hardware that I included in my build-a-box. The focus of this course was on networking so due to the smooth nature of the physical system build, relatively little time was devoted to the intricacies of modern hardware configurations. I then paused to read a book on networking fundamentals to support my understanding when I was actually setting up a network with Linux. I then dug into the Linux-specific implementations of various Internet protocols. Finally, with my routers setup and functioning, I installed and configured a LAMP stack on the machine I built and configured it to serve a Moodle site to the WWW. The next three sections of this report will elaborate on each of the core dimensions of the project:

1. Hardware acquisition, assembly, and CentOS7 install.
2. Swapping out router firmware and setting up my personal LAN.
3. Configuring the LAMP stack on the build-a-box to serve a Moodle site to the Internet.

Hardware acquisition and CentOS 7 Install

As someone relatively new to hardware, I was amazed that I had zero issues getting the box built and the OS installed. I read about 30 tutorials and guides on choosing Linux-compatible hardware, ordered the parts from Amazon, and assembled them using the motherboard's install guide--which was very detailed. The following table lists the core components of the box:

HW Component	Considerations	
Processor	AMD AMD A8 7600 FM2+ 4MB Box R7 Series Graphics 3.8 4 Socket FM2+	The A-series is a middle line series, noted for much better performance per dollar than the A4 and A-6 series from AMD. I decided to go with the AMD with integrated graphics because I've never worked with an AMD chipset before and I generally like to use hardware that is compatible with many different brands. (Intel i understand has more rigid specifications on motherboard, etc.) The integrated graphics card is all i need for this server, which might even

		be running headless. Also, it is popular and mainstream so has plenty of linux support. The FM2+ socket seems widely supported in the industry now, even if it is on the newer side. I also read that one can tweak the resting power usage, which will be great since I think this box will be on all the time.
Motherboard	Gigabyte FM2+/FM2 AMD A78 HDMI Dual-Link DVI D-Sub 2-Way Crossfire mATX Motherboard GA-F2A78M-D3H	This motherboard fits the processor and has 6 SATA heads and integrated ethernet. Seems very functional.
Power Supply	Senty 550W 80 plus bronze	I read that 550-watt is plenty for a most Pcs that don't have a fancy graphics card, which I won't be getting. It has 6 SATA plus and the 24-pin for the motherboard
Case	Sentey® KRON GS-6005 Desktop Gaming Computer Case	This case looks cool and is cheap wiht good reviews, compatible with my mATX motherboard (Specs cont: / USB 3.0 + 3 x USB 2.0 / HD Audio + Mic / Card Reader and Micro Sd Incluided / Side Panel Transparent Window / 330mm VGA Length Support / 135mm CPU Height Support / 120mm Front Blue LED Fan Cooler + 120mm Rear Blue LED Fan Cooler / ATX & M-ATX Support)
Hard Drive	WD Green 2TB Desktop Hard Drive: 3.5-inch, SATA 6 Gb/s, IntelliPower, 64MB Cache WD20EZRX	Highly rated, w/linux support. This drive can hold files of all kinds which will be handy since I'd like this box to be my main file server.
	Samsung 850 EVO 250GB 2.5-Inch SATA III Internal SSD (MZ-75E250B/AM)	I've never run an SSD before, and this one is widely used, linux compat, and pretty cheap. I'll install the OS on here, of course. Well-reviewed and decently fast.
Ram	Crucial Ballistix Sport 8GB Kit (4Gbx2) DDR3 1600 (PC3-12800) 240-Pin UDIMM Memory	Full compatibility with the motherboard

For about $400 this was a decent machine that will allow for lots of tinkering and expansion. I setup a RAID array on the SSD and larger magnetic drive that will allow for seamless logical partition expansion. I may want to turn this box into a LAN file server and replace the LAMP server with a VPS like Inode or Digital Ocean. It's obviously not a fast machine or for gaming and I will probably reinstall and run it headless once I'm comfortable that I don't want to play around with tools on it that also have GUI components that I'd like to learn about in parallel to the command line options.

Hardware-related resources

I found a few key sites very helpful in planning for and assembling the box:

http://www.kitchentablecomputers.com/linux3.php - A general Linux-focused hardware discussion site that seems to be driven by one person or small group of dedicated folks.

http://www.tomshardware.com/ - This seem to be the oft-referenced site when one is looking on forums for advice on hardware choice.

http://pcpartpicker.com/ - User-driven site around individual boxes built. This was very handy to use to compare the trade-offs that folks were making when thinking about a “cheap box” that does “this and this”. I didn't copy any box on here, of course, but it was nice to check compatibility for a given part I was looking at since the users would write the OS they installed and often make useful notes about their own HW choices.

http://www.geeks.com/techtips/2009/techtips-29MAR09.htm - geeks.com had a great tutorial on motherboards that, much like the motherboard in a computer, formed the foundation of the considerations for the rest of the parts in the box.

[I ended up buying the parts from Amazon since I'm a prime member and it's convenient. I felt a but scrungy though, being so pro-open source and then buying from a big far away company instead of a local computing store. I went to a few repair shops and tried to ask folks if they personally knew any parts stores that had folks willing to help customers. I didn't get any suggestions, and it was in the middle of the term, so I just had them dropped on the doorstep. Thanks to the UPS and FEDex folks who got the boxes here without a single HW related install issue.]

Here are a few images and outputs with some basic HW configuration printouts.

Disk drive configuration

Here is the human readable disk setup for the box:

```
[ecds@centserv Pictures]$ df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/centos-root00  47G  244M   47G   1% /
devtmpfs                  3.3G     0  3.3G   0% /dev
tmpfs                     3.3G   16M  3.3G   1% /dev/shm
tmpfs                     3.3G  9.1M  3.3G   1% /run
tmpfs                     3.3G     0  3.3G   0% /sys/fs/cgroup
/dev/mapper/centos-usr     94G  5.4G   88G   6% /usr
```

```
/dev/sda2                497M  304M  194M  62% /boot
/dev/sda1                200M  9.5M  191M   5% /boot/efi
/dev/mapper/centos-home  187G  244M  186G   1% /home
/dev/mapper/centos-var    94G  1.1G   93G   2% /var
tmpfs                    670M   20K  670M   1% /run/user/1000
```

As you can see, not much of the 2.25 TB shows up on this disk output. I deliberately decided to use logical volume mapping so I can expand into the now un-allocated space with new partitions for other experiments with other OS's or storage systems. I also might want to try booting off of only the SSD. Now, my RAID conditions are such that the data to run the system is scattered on both drives, so my supposition is that I'm not getting the super fast read time of the SSD because I'm accessing the magnetic disk so frequently for core system tasks.

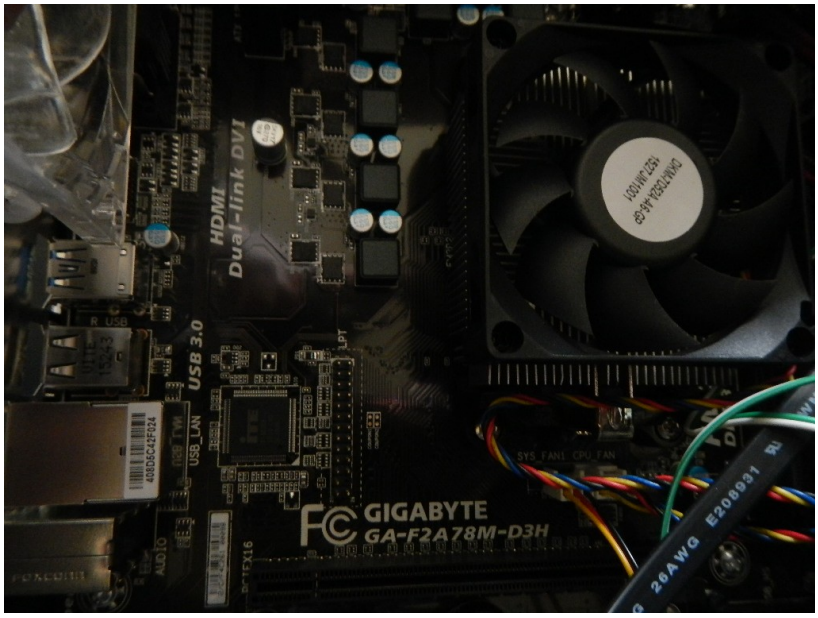
Photos of the home build

Here are a few shots of the actual build finished product! The case even has cool blue LED lighting inside and a transparent side panel (which faces my wall...).

The Uninterrupted Power Supply unit all plugged in an ready. I think there is a bug in this guy, though, since it cycles the whole strip off and on whenever I print anything (even with the printer not plugged into the UPS itself—it still cuts out with the printer on the same breaker). The theory is that the printer spikes in its current draw for about a half second upon activation. The UPS might be good at switching to backup power seamlsesly but it doesn't seem to correctly switch back to wall power without a cutout. Clearly, I can't have my routers power cycle every time I print! Bugs to work out.

The tower awaiting its brain!

The guts: the 550 w power supply is visible. The two HDDs are in the rack on the right. The motherboard has integrated networking and graphics cards, so there aren't any extras plugged into the motherboard. Only SATA cables.



You can make out the manufacturer of the motherboard in this shot, plus a nice view of the CPU fan and an empty RAM slot.

Foundational networking reading + LAN Setup

With my CentOS 7 server all ready to serve, I needed to know how to connect the cables and configure the routers. I started by reading what I read was a decent introduction to the core concepts of the OSI networking model.

- Elements of Computer Networking: An Integrated Approach (Concepts, Problems and Interview Questions) Authors: Karumanchi, Narasimha, A, Dr Damodaram, M, Dr Sreenivasa Rao

This great text is platform neutral and instead focuses on the nuances of routing packets and the hardware to do so efficiently from a computer science standpoint. Since I anticipate spending most of my time in the upper OSI layers, I did skip a careful read of the chapter on routing algorithm comparison. Skimming the headers of the skipped chapters did help to give context to the kinds of issues that folks think about who are working down on the IP layers.

I also purchased these two books published by O'Reilley that I referred to as needed while I setup my network. The basic 5 chapters of the Administrator's Guide did an excellent job of teaching me how to read routing tables and how to understand gateways, routers, addressing, subnets in the Linux context. I started working alongside the authors as they talked through the specific configurations for a similarly simple LAN with a few subnets.

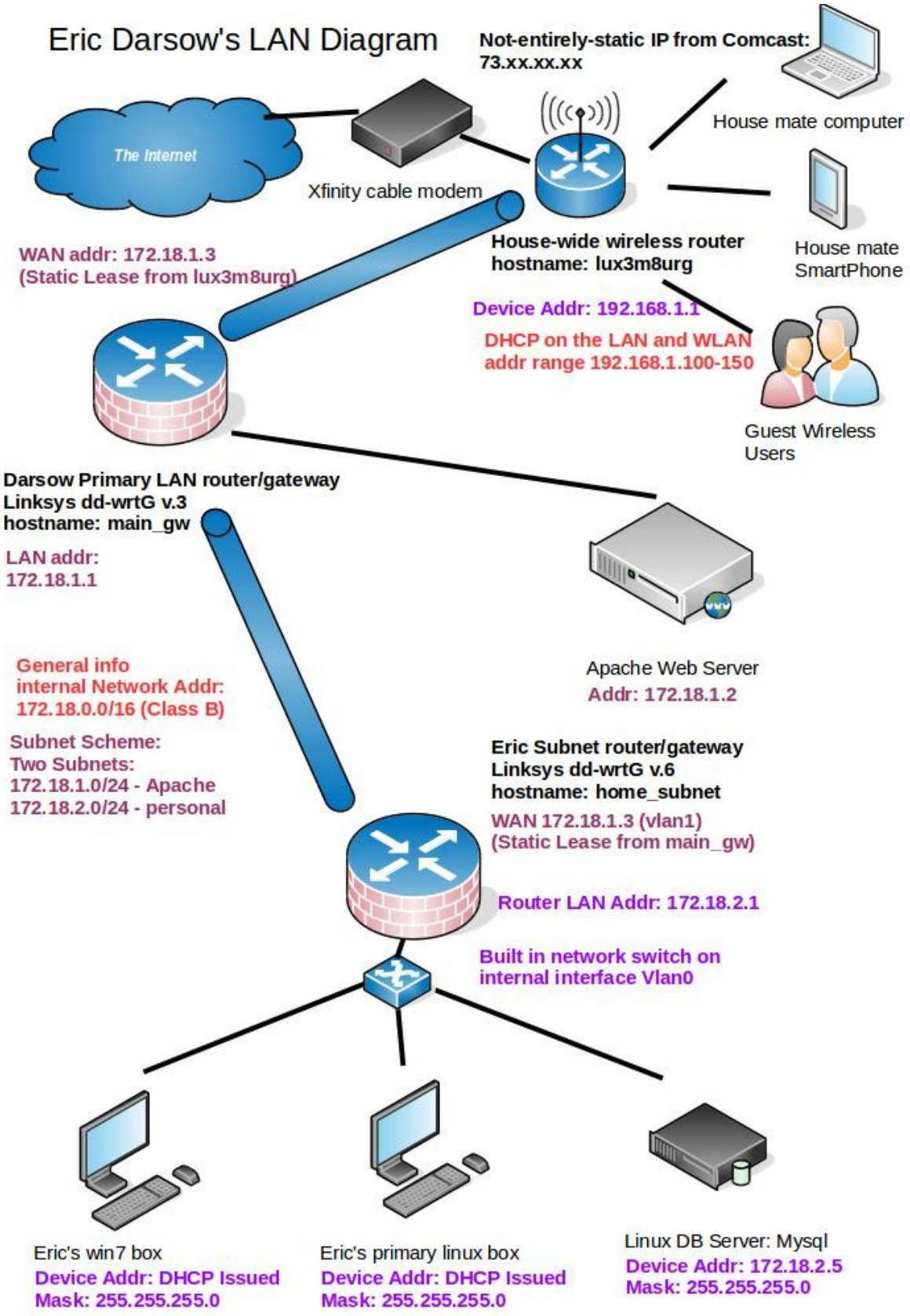
- Linux Network Administrator's Guide Authors: Bautts, Tony, Dawson, Terry, Purdy, Gregor N.
- Linux Networking Cookbook Linux Authors: Schroder, Carla

With consideration with Professor Moul, I settled on the following conceptual setup of the home LAN:

- Maintain a stable gateway into the *house* (which will be used by me and my house mate + house guests). This gateway is also the wireless LAN access point and the only functioning WAP in the house. This router is a Netgear WNDR3400 and came with DD-WRT installed! I read good things about Negtear on the open source routing forums, particularly for Open-WRT, and was pleasantly surprised to see DD-WRT come installed from the OEM.
- This main GW functions as the firewall between our house LAN and the Internet. I then have my own Subnet which is the 172.18.1.0 network. The LAMP server is on this subnet.
- I also setup another subnet on 172.18.2.0 so I can run hosts behind a firewall downstream from the APACHE server for added security and so I can play around with a router that the server isn't attached to.

These concepts are implemented in the my home LAN represented by the schematic on the following page. One can think of the downward traffic from the home router down to the 'eric_subnet' as proceeding with decreasing failure impact. Since it's harder for downstream hosts to affect upstream ones, it made sense to make my core playzone as safe as possible.

Home LAN Diagram with AP addresses and hostnames



Notes on network setup

My original intent had been to serve the Moodle site off of a Linux/apache/php server in the 172.18.1.0 network *from* a server on the downstream 172.18.2.0 subnet. I could get upstream pings just fine but I couldn't get the connection to work from the 172.18.1.0 essentially *over* to the subnet 172.18.2.0 which was attached to a normal LAN port on main_gw. This could be done with VLAN trunking and I spent about 5 hours trying to configure a new interface on main_gw to do so, a virtual one, and then attempted to route traffic to it by editing the main routing table. I could get ifconfig to show me the new vlan2 interface and I thought I had correctly mapped it to an actual port number on the box but I couldn't get the packets to make it down/over to the lower subnet. In the grand scheme of the many tasks that had to be learned, this nuance of creating virtual LANS via the command line would have to wait.

Configuring the port mapping to get packets bound for the LAMP server from the Internet was mechanically not very difficult. However, as the process of learning goes, I first attempted to tell the house's gateway, lux3m8urg, about the subnet 172.18.1.0 existing and asking it to help with the routing from the Internet. Then I realized that I just needed to tell lux3m8urg that any incoming packets from new connections on port 80 need to go to the host that it gave a static DHCP address to on subnet 192.168.1.0—the main_gw! Once the packets got to my main_gw, I could tell main_gw to forward any packets incoming on port 80 to go to the server on the static lease 172.18.1.2.

Hardware choice considerations

The choice of hardware for each of my three routing points was intentional. The Linksys WRT54G/S series is a great one for us Linux folks due to its flexible NVRAM configurations and high cost-to-tinker value. The Tomato router folks swear by the model and many folks speak of the *good old days* before the WRT54G was “updated” to version 5 which only had 2 mb of NVRAM and could not run the full version of DD-WRT as a result. Having familiarized myself with the downgrade in internals prior to buying the devices, I obviously intended to get two version 4 or earlier WRT54G routers. Yet, a misleading ebay ad left me with one desired version 3 and one inadequate version 6. Setting aside the version 6, I installed the newest Tomoato firmware release on the solid WRT54G v.3 which became the main_gw. This router is the gateway for the LAMP server. I left the OEM installed DD-WRT on the house's main gateway and WAP since its firewall is doing all the heavy lifting and I thought it to be a Good Thing that I didn't have a chance to miscalibrate some important iptables setting and expose the house to all sorts of Internet riffraff inadvertently.

The question then became what to do with this whimpy little version 6 that I coudln't put Tomato on because it's memory is too small. I decided that I was up for the task of trying a DD-WRT *mico* install on the version 6 since the forum posters alleged that it *can* run satisfactorily (without any extra modules) in about 1.2 mb. I followed a great tutorial on how to prepare for the flashing, wipe out the current firmware in a very complete way and then use binary ftp to hurl the new firmware at the little emtpy box that basically only knows how to listen for some very simple commands.

https://bitsum.com/openwiking/owbase/WRT54G5_CFE/#h10

I was so impressed with the man who made these special files for doing this on just the version 5 and 6 of the WRT54G that I'm going to give him a donation. He embodies the kind of tinkering and sharing spirit that made this whole project worthwhile on many non-technical dimensions.

This wikipedia entry on open firmware projects was a lifesaver as I was getting overwhelmed with people talking about DD-WRT, Open-WRT, TOMATO, and all the forks off of them.

https://en.wikipedia.org/wiki/List_of_router_firmware_projects

And finally this InfoWorld posting was very through in actually giving really solid context about what is going on with these various projects and approaches to open routing:

http://www.infoworld.com/article/2607851/networking/networking-teach-your-router-new-tricks-with-dd-wrt-or-openwrt.html

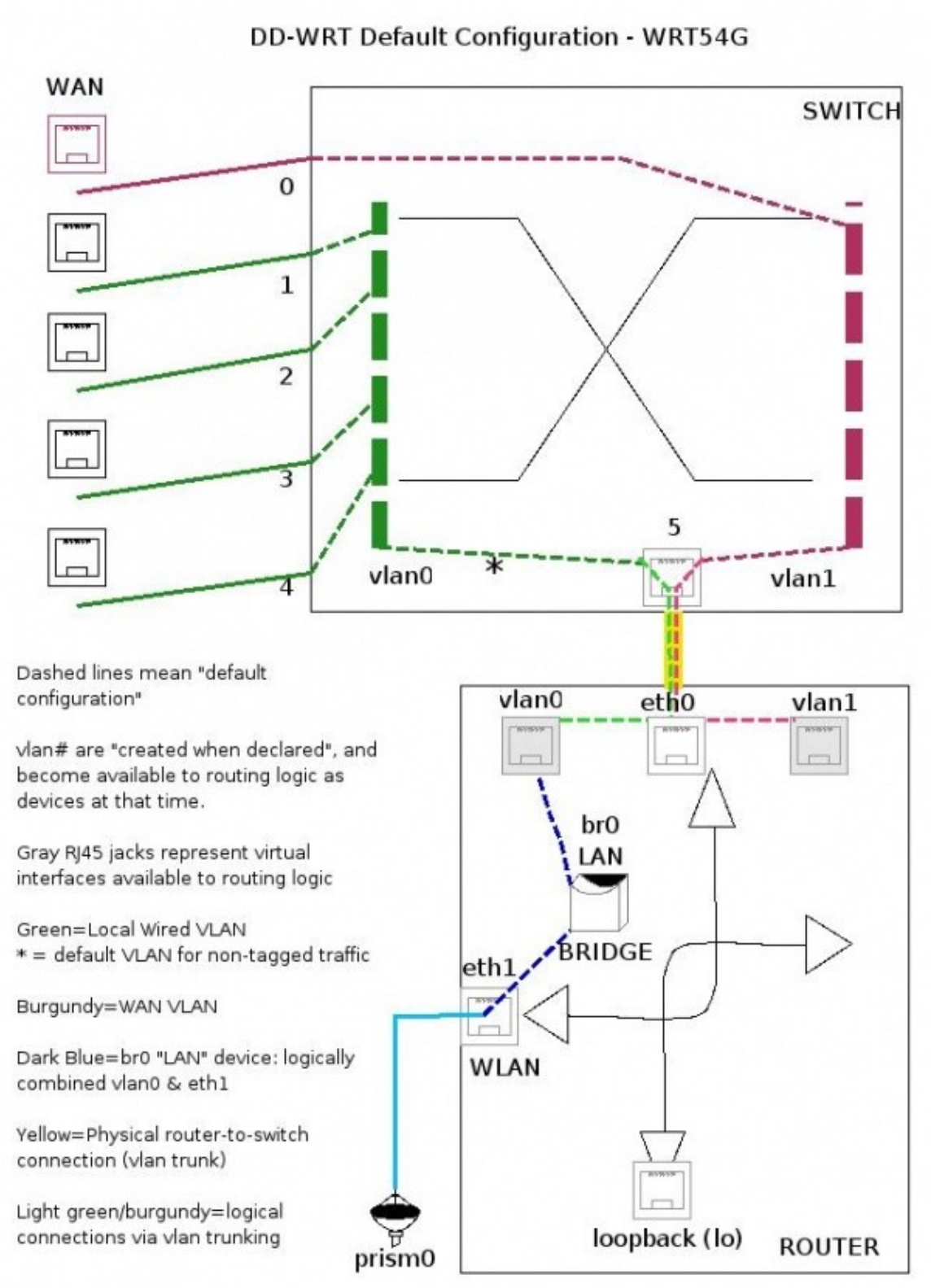
Obviously Open-WRT, Tomato, and the Linksys forums were instrumental in figuring out how to work on the command line to setup all of the routing updates for each of these systems. I did use the GUI for the house router, once again, to avoid putting the house at the whims of my *noobness.*

My three routers, all running linux-based firmware. The names and addresses on the colorful stickies correspond to the network schematic shown above. The bottom 'play with me' router will get an install of Open-WRT so I can tinker with the many modules offered!

With firmware installed on my new routers (pictured above), I set out to configure the devices using the variety of Linux boxes I've accumulated. My intent was to work on basic routing in each subnet to familiarize myself with reading and adjusting routing tables via SSH sessions on each router. Once I could configure DHCP correctly within the 2 subnets, I then connected them and setup static leases, etc. to the routers, from the rotuers. Doing so required becoming familiar with the internal architecture of the Linksys WRT54G, particularly which interfaces were used to interact with which ports and which conceptual network structures. Such a simple diagram clarified my understanding of how a router box like the WRT54G has several “ip address” depending on which interface one is talking about. By way of an additional example, understanding what an interface like br0 does is not at all intuitive from looking at the device's exterior ports and notes but

when looking at the below schematic, one can easily see that this is a bridge between the WLAN and the LAN. This was so helpful that I posted this schematic of the DDWRT V4 above my workstation and referred to it continuously as I worked on my IP tables and route commands.

DD-WRT Interface and port configuration

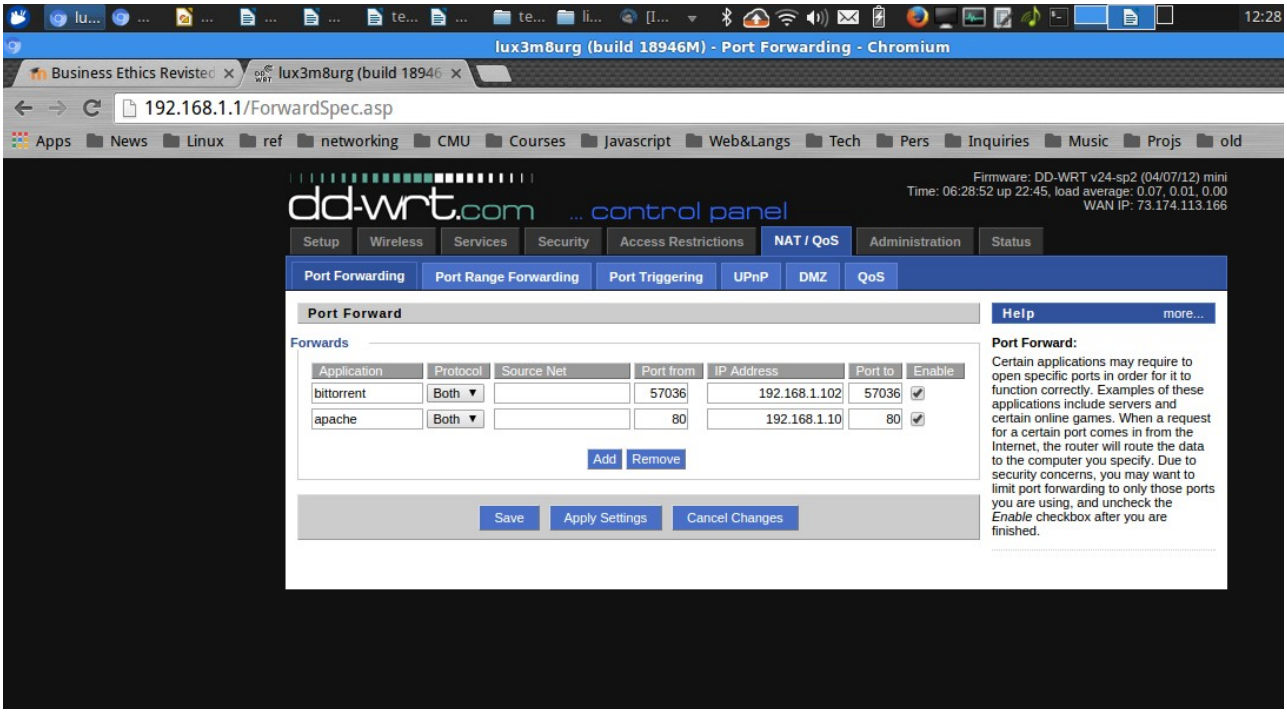


A very useful schematic of the Linksys WRT54G V3 and 4

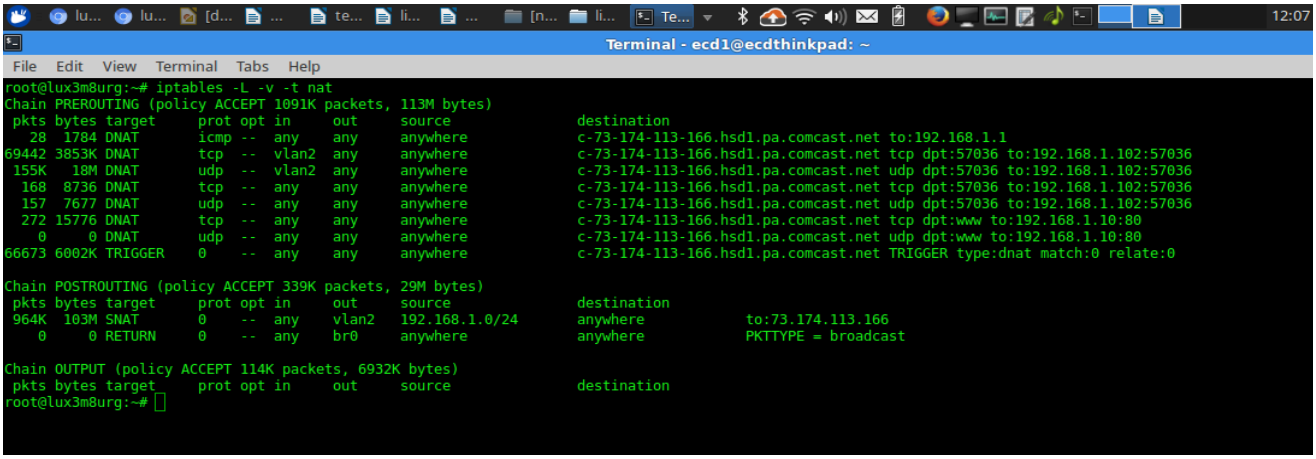
Router configuration explanation and screen shots

After about two days of reading and tinkering with the new firmware, I had my devices setup. Here are some screen shots of the router GUIs and the ssh sessions displaying core features that I found useful in my learning. I'll highlight and comment on various features that I configured such that there's at least one screen of output from each of the three configured routers (refer to the network diagram above for the names and address translations relevant to these screen shots).

Starting at the house router and working downstream, this is the routing table for the house router called lux3m8urg and we can see the routing table as shown with the GUI as well as the output of iptables -L -I shown via SSH session:



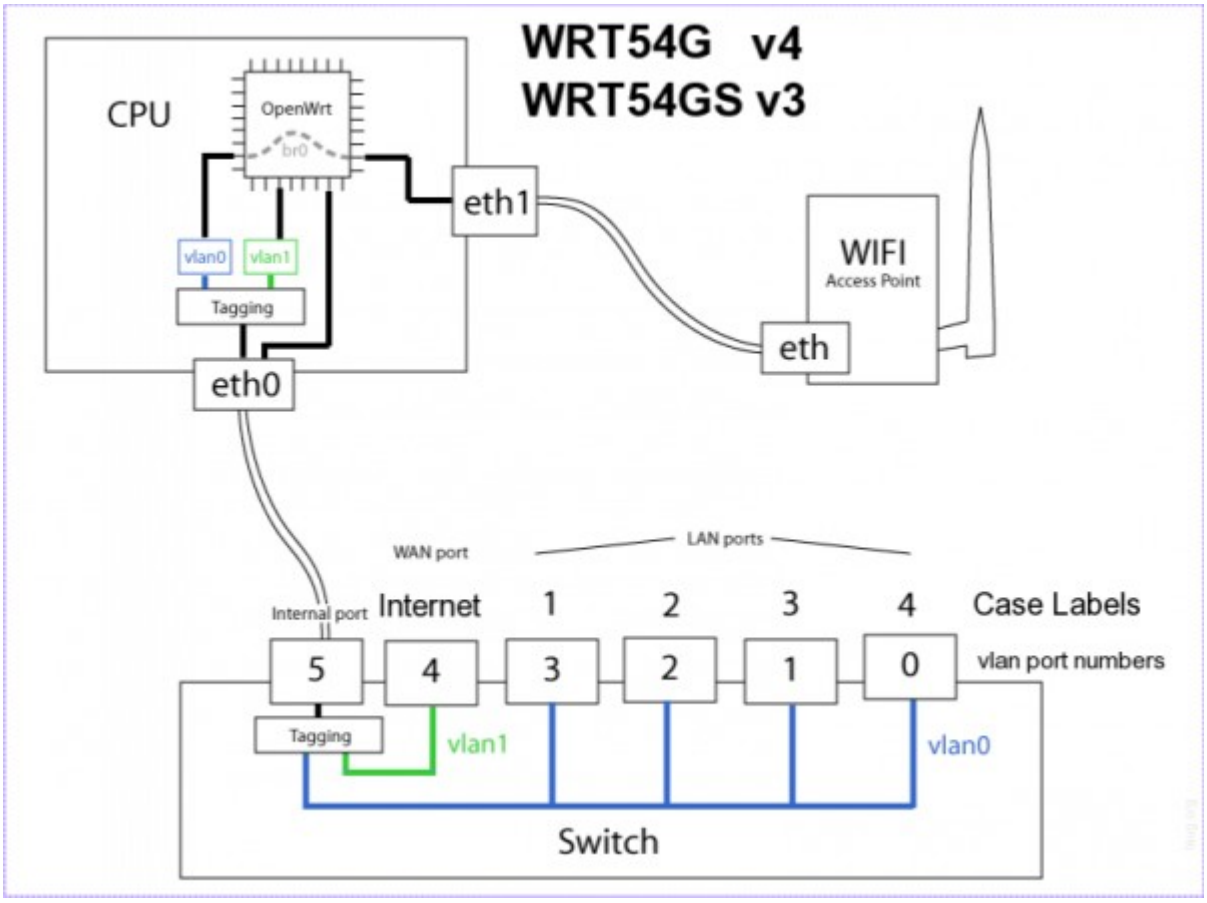
Port forwarding on the house_router: traffic on torrent ports go to housemate, and HTTP traffic are routed to my CentOS Server. I will add HTTPS on port 443 when I get the apache modules installed.This GUI adds entries to the filter and the NAT table (DNAT) automatically! So handy!



The NAT table on the house_router with interfaces shown.

My housemate has some bittorrent ports forwarded to his computer on the WLAN and my port 80 is making it to the host with the static IP 192.168.1.10 which is leased to the main_gw's WAN port. Even though I used the GUI for this routing table, this was not my first approach; I wanted to know how the GUI was interacting with the Kernel, so I was constantly reviewing the iptables list outputs to see how the GUI changes are reflected in the underlying netfilter configurations. After several hours of careful viewing along with referencing the Sobell text on firewalls, I discovered that examining ipables rules could be much more instructive with the -v option that lists the interfaces that each rule applies to.

With the interfaces of the WRT54G listed, I could also begin to see the potential differences in internal architecture between the Linksys models and the Negear WNR3400. I couldn't find a schematic of this Netgear device after about 15 mins of searching, so curious comparisons had to suffice. Even with only the Linksys schematic, there seems to be enough overlap between the models that understandings any router insides thoroughly suddenly gives birth to powerful moments of clarity surrounding how the routing and firewalling process unfolds. Here is another useful schematic of the WRT54G that represents a less detailed view of the system which complements the above diagram.



Clearly, `netfilter` and it's `iptables` front end tool are at the heart of Linux security. As a result, I read a number of discussions about the merits of using other tools aside from iptables for firewall configurations, such as GNU's Uncomplicated Firewall or the standard on CentOS 7 tool which is called `firewall-cmd`. In the end, one smart sounding poster was persuasive in saying that one really needs to learn IP tables and figure out how to read the tables correctly and edit them. His review of various iptables helper tools suggested that they aren't all that much more useful and tend to remove users from really understanding what the kernel is doing with its packets. This was a persuasive argument, so into iptables I dove.

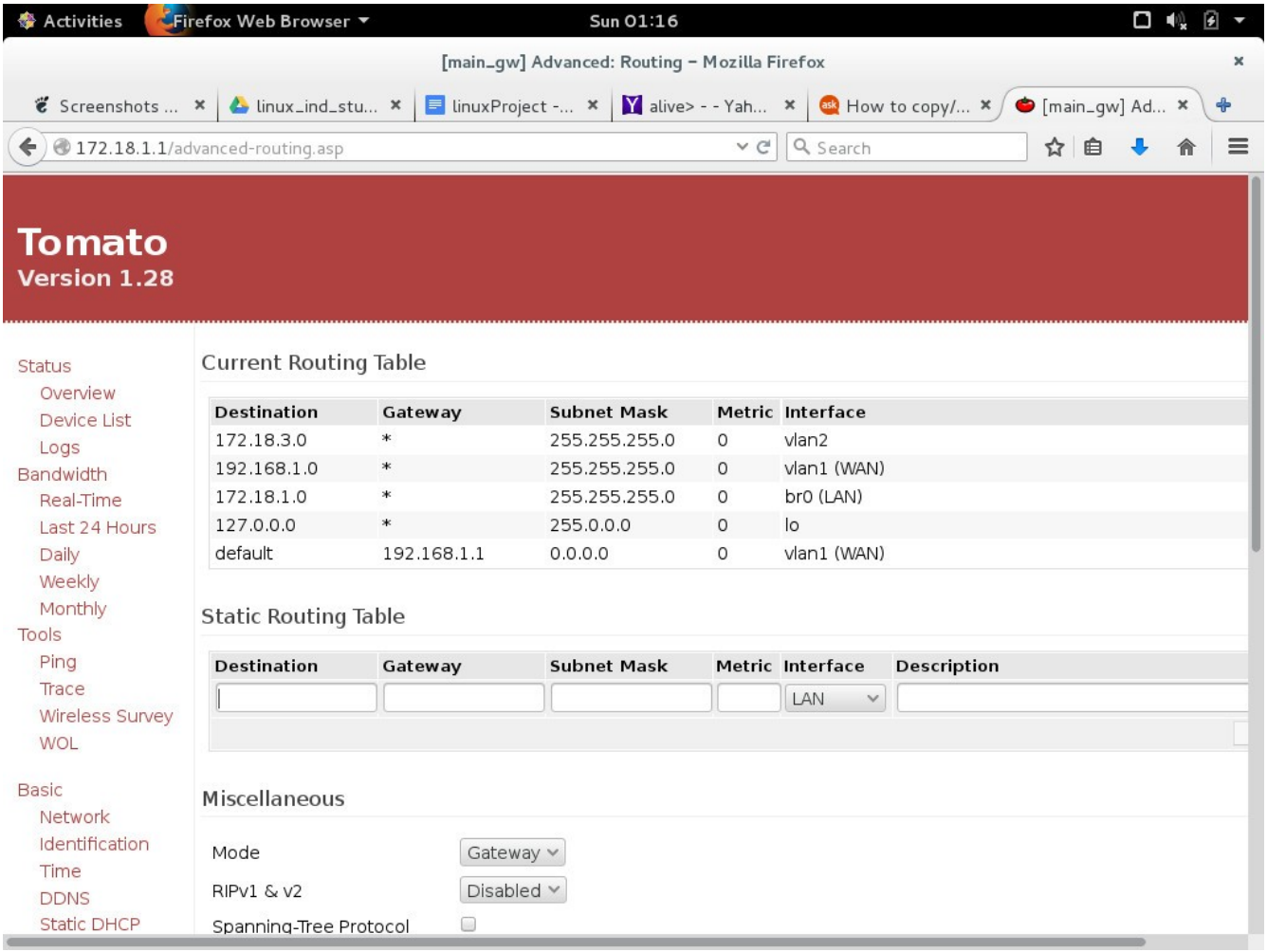
This a diagram of the core concepts of `iptables` that I compiled from reading the Sobell and looking at a few influential but still muddled conceptual diagrams of the relationship between chains, tables, and targets. This diagram is best to be accompanied buy a routing diagram that shows the sequential relationship between these entities. Having scratched my head for a long time to understand what is actually going on with what kinds of packets, I am convinced that one should examine and reflect at least two conceptual/visual approaches to `iptables` firewalling.

Iptable firewall relationships: chains, rules, tables, and targets

<table>
<tr><td>TABLE name</td><td>NAT</td><td>MANGLE</td><td>FILTER</td></tr>
<tr><td>Description</td><td>Change source and destination addresses and ports for hiding and sharing.</td><td>Alter the TOS or TTL and MARK attributes in the packet headers.</td><td>Drop or accept packets but no altering!</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="5">Built-in chains (of rules) in each table.</td><td colspan="2">PREROUTING</td><td></td></tr>
<tr><td colspan="3">OUTPUT</td></tr>
<tr><td></td><td colspan="2">INPUT</td></tr>
<tr><td colspan="2">POSTROUTING</td><td></td></tr>
<tr><td></td><td colspan="2">FORWARD</td></tr>
<tr><td rowspan="2">Applicable targets</td><td>DNAT, SNAT, MASQUERADE</td><td></td><td>DROP, ACCEPT, REJECT (tell sender)</td></tr>
<tr><td colspan="3">RETURN, LOG (all)</td></tr>
</table>

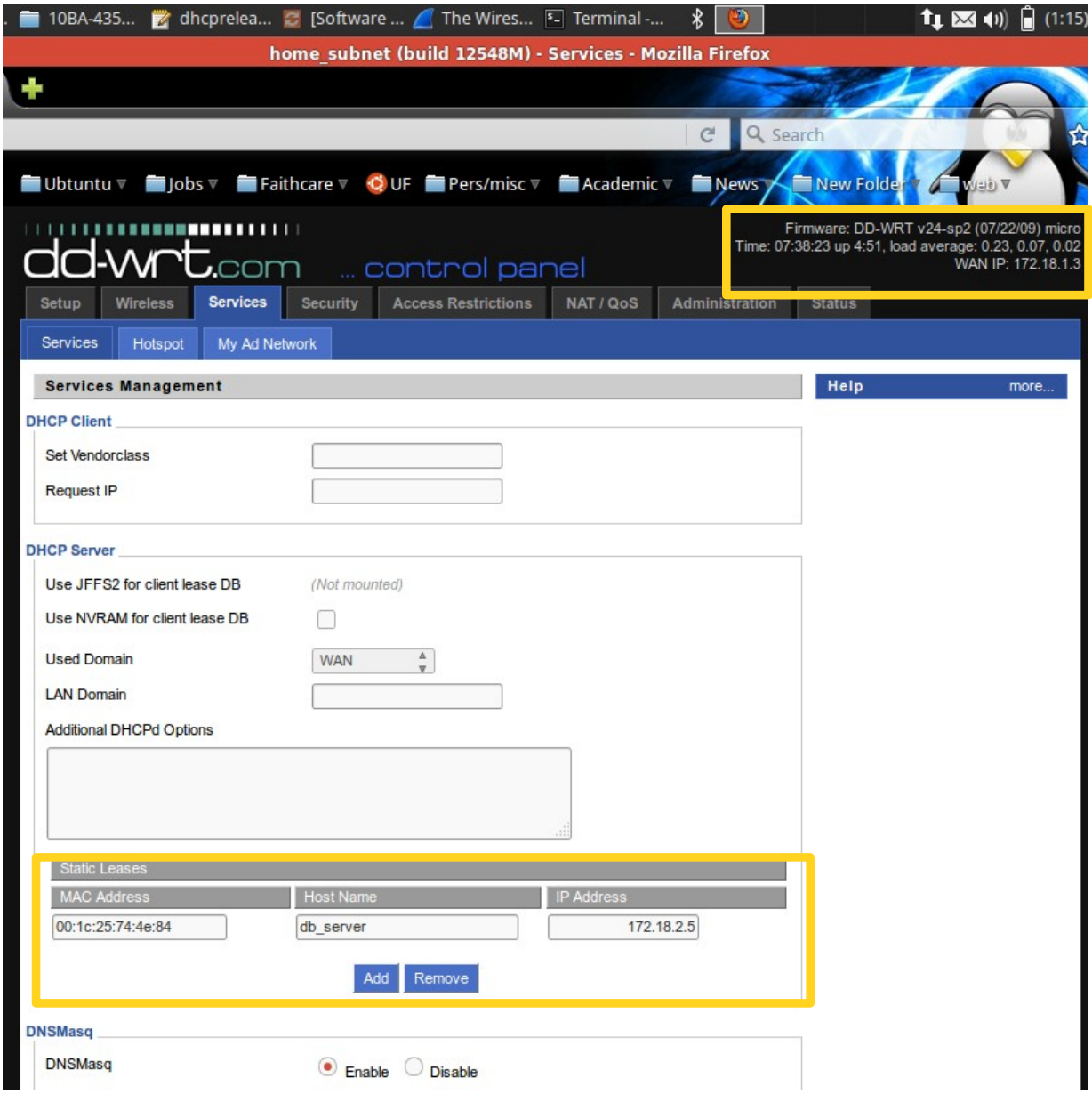
This table may have technical flaws, and I invite the conversations that correcting could bring. I share the table to illustrate a feature of my learning process throughout this course: each individual tool or framework (e.g. SELinux) required a very engaging set of conceptual understandings to congeal mentally in order for the words on the console to make sense. Scribbling and adjusting diagrams like the one above *by hand* while reading and poking at the command line was a central part of the learning experience. It helped me read dense reference and tutorial texts. As my familiarity with a given tool increased, I was also also able to use my emerging conceptual understanding to spot moments in which quickly written blog posts are glossing over details and relationships that are important to grasp for a given tool, like `iptables`, but not particularly intuitive to grasp and would require more explanation.

Continuing the router-based descent into the network, the image below shows the configuration screen of the Tomato router functioning as main_gw in the network schematic:



A screen shot of the routing table for the main_gw. I still have an entry for a network 172.18.3.0 that I hope to configure in the future

Finally, this is the micro version of DD-WRT running on the Linksys WRT54G v.6—the one without enough NVRAM for DD-WRT *mini* which is the standard release. One key feature that is missing in the smaller binary is the important SSH daemon; it only supports telnet out of the box. So clearly this would not be a useful router to run in critical infrastructure points as it cannot be accessed securely from a remote location (that is, without one needing to be tricky and careful about SSHing into another device on the subnet securly and then `telnet`ting into the DD-WRT micro device locally).



This is the <2mb micro version of DD-WRT installed on eric_subnet. You can see in the upper right corner the version is 'micro' and the WAN IP address is the static address served to it by main_gw. Finally this config screen shows the static lease given to my linux db server on the 172.18.2.0 subnet.

DHCP Demonstration Output

I became very familiar with the DHCP leasing process through all the tinkering. By way of demonstration, following this paragraph is a formatted script of a BASH session on the system that was destined to be the DB server to server the LA_P stack server. First, it is plugged into a LAN port on the main_gw (172.18.1.0 network) which gives it a static lease of 172.18.1.4. Then I physically changed over the CAT5 cable to plug into a LAN port on the home_subnet router (172.18.2.0 network). Since my IP configurations are all handled by the router's DHCP configurations and not via interface-level configuration on each individual host, one can see how after I asked the ethernet interface on the computer to release and renew its IP address after the cable change, interface eth0 is given the assigned static IP address of 172.18.2.5 on 172.18.2.0 by the eric_subnet router as planned on my network planning schematic above. Note that `dhclient -r` releases the lease and leaves the interface address-less. A simple dhclient eth0 prompts the lease request process resulting in a new address. Cool!

```
christophe@christophe-ThinkPad-T61:/etc$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1c:25:74:4e:84
          inet addr:172.18.1.4  Bcast:172.18.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:25ff:fe74:4e84/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1907 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1863 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2740449 (2.7 MB)  TX bytes:169439 (169.4 KB)
          Interrupt:20 Memory:fe000000-fe020000
```

... [loopback interface removed]

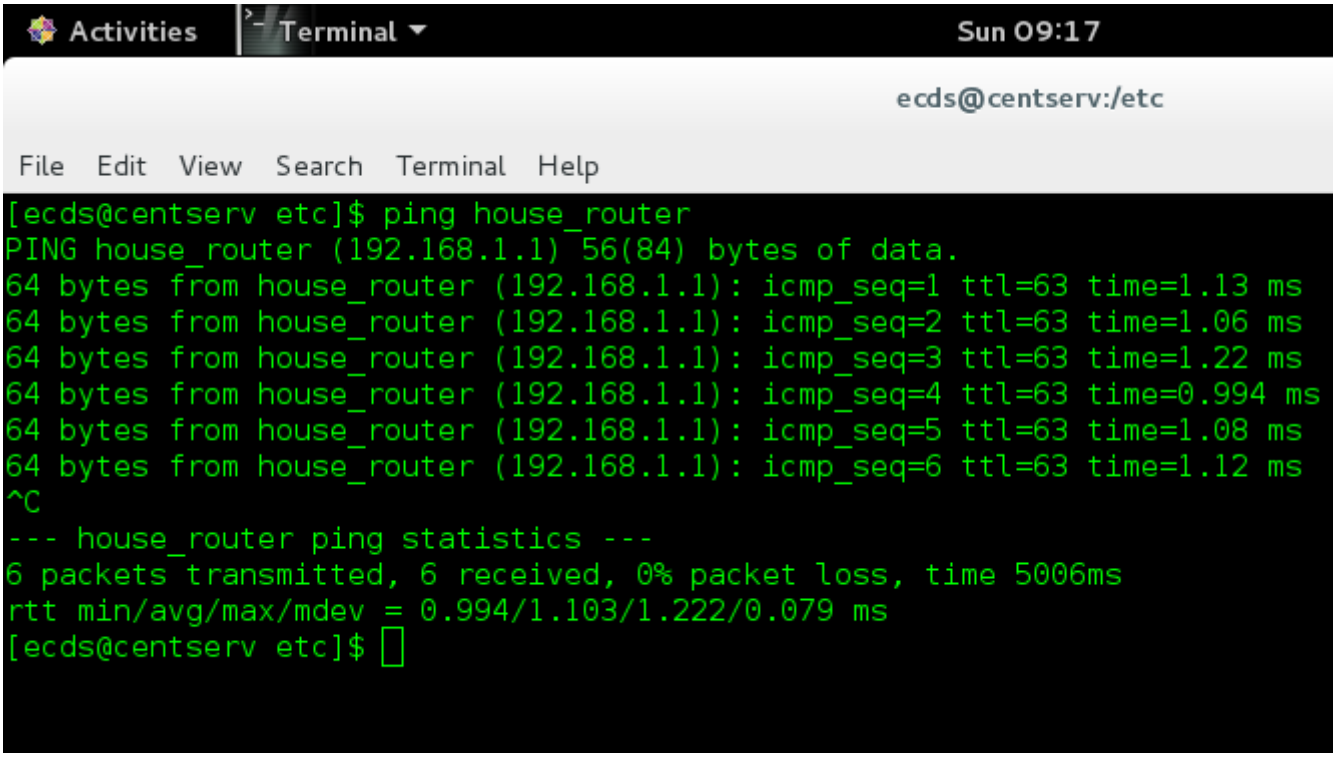
```
christophe@christophe-ThinkPad-T61:/etc$ sudo dhclient -r eth0
christophe@christophe-ThinkPad-T61:/etc$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1c:25:74:4e:84
          inet6 addr: fe80::21c:25ff:fe74:4e84/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1907 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1866 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2740449 (2.7 MB)  TX bytes:169631 (169.6 KB)
          Interrupt:20 Memory:fe000000-fe020000
```

... [loopback interface removed]

```
christophe@christophe-ThinkPad-T61:/etc$ sudo dhclient eth0
christophe@christophe-ThinkPad-T61:/etc$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1c:25:74:4e:84
          inet addr:172.18.2.5  Bcast:172.18.2.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:25ff:fe74:4e84/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1915 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1887 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2741855 (2.7 MB)  TX bytes:173973 (173.9 KB)
          Interrupt:20 Memory:fe000000-fe020000
```

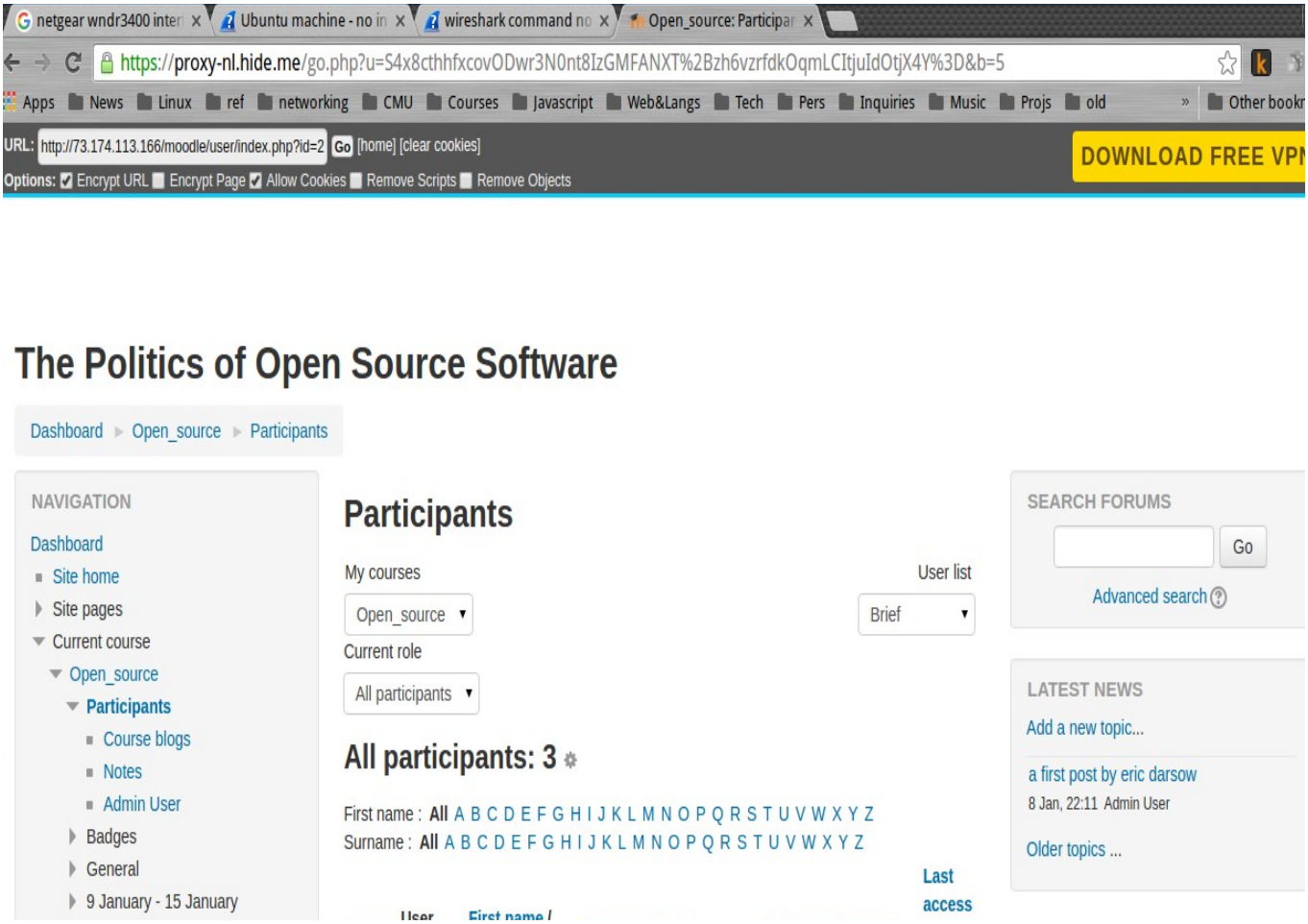
Demonstration of /etc/hosts

I haven't configure a domain name server for just the LAN yet, but I did edit /etc/hosts files on each of the boxes so I could use easy English names for pinging and SSHing into various devices. Here's a screenshot on my CentOS server of me pinging the house-wide router from inside the main_gw.



LAMP software stack configuration and Moodle installed

My end goal was to serve a moodle site to the Internet from the CentOS server. I successfully did so after a relatively minimal amount of hassle. While most of the configuration directives for Apache are in decent shape out of the box, they are insufficient for allowing an application like Moodle to access a data directory outside of the document root for Apache. I decided to use the directory /opt/moodle for this purpose. I gave the apache user ownership and the Apache group rights to this directory and adjusted the context type of that directory to avoid SELinux access errors. Here is a screenshot of the Moodle site as viewed from an arbitrary website driven proxy server pointed to my LAN's gateway using the xfinity-issued public IP address. Success!



This (somewhat ugly) screen shot is the culmination of the project! This is a window in a web proxy that is pointing to my public ip address (see the form field in the upper left). The page loaded is the participant list of the first moodle course I created on my Moodle app hosted on the CentOS Server.

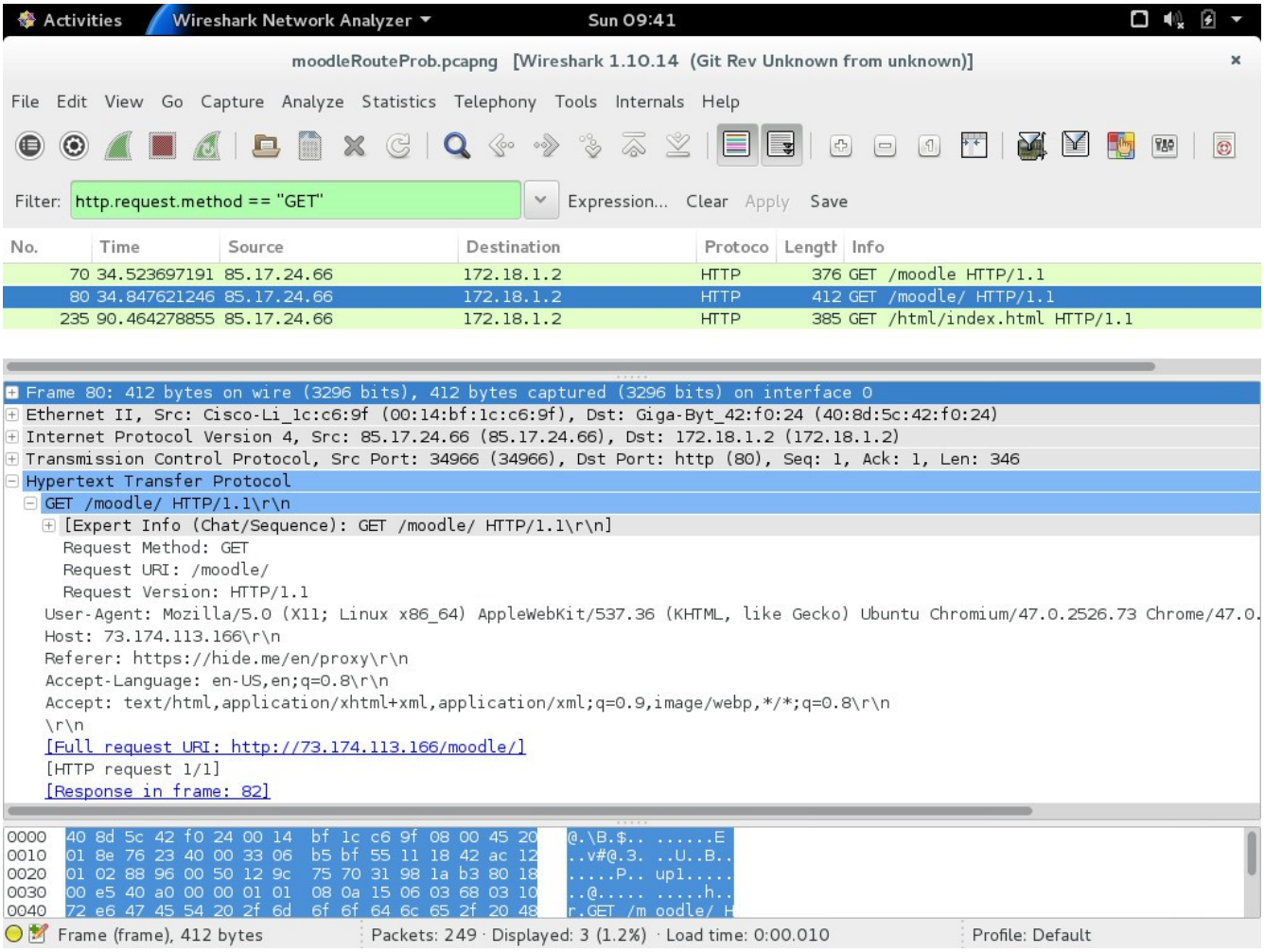
Moodle configuration discussion

The hickups in the Moodle install process were related to nonfunctional database user access and the format of the request URL for the application. The database access issue seemed to stem from a mistyped password in the /etc/httpd/moodle/php.ini configuration file. The Moodle installer could connect to the MySQL database but the user didn't have proper table CREATE privileges. I re-read the tutorial where I thought I had given the moodluser such rights and retried the command after cross-referencing my MySQL reference book.

The second issue concerned the exact URL used to access moodle. Also in its php configuration file, a specific string is checked against the request URL coming into Apache. If they don't match perfectly, Moodle doesn't serve any content. This process is error-prone such that my first attempts to load moodle via the loopback failed because I hadn't typed 'http:// before the static LAN IP address for the server (172.18.1.2). I thought the browser would append this before sending the GET request, but it doesn't do so reliably, leading to a refusal to load error. The Moodle app does have a redirect mechanism to help with this picky behavior but it was not working properly (probably another PHP config issue).

After I could get to Moodle from a host in the main_gw subnet, I decided to take it live to the Internet. I was preplexed because I could access other content in my document root directory from the Internet but the moodle site was giving me the same error regarding the access string. I thought to myself at first: well, I don't know what the request string will be when the GET reaches Apache since the incoming request is adjusted via the NAT tables in the house's firewall two hop above the server host.

To check exactly what Apache sees in a packet that has reached the inside my LAN from a proxy server in the Internet, I used Wireshark on the server to sniff all incoming packets. I set the system to sniff while sending the GET request from the proxy server (accessed via my personal computer connected to the Internet through the house router's Wireless LAN). I saw the GET packet pop up and I looked into the application layer of the header and, sure enough, right there is the http://73.174.xx.xx/moodle URL intact. When I saw this, I changed the Moodle PHP config file to include not the LAN address but my assigned puplic IP. After a restart, I could successfully access moodle from the Internet. I will, of course, need to setup a Dynamic DNS server for my home server and figure out what to do with the moodle access string, but this detail is secondary to the excitement of having used a powerful network tool to debug the routing issue. Here's a screen shot of the GET packet in Wireshark coming in from the Proxy.



Wireshark session output from a sniff on the CentOS server as the proxy sends at GET request.This was useful in debugging a Moodle configuration file.

I was thrilled to see this packet preserve the request header from the browser because it confirmed what I had not consciously connected which was the fact that the router's Network Address Translation process is an IP-layer interaction; the application layer header that contains the text of the GET request's URL is untouched in the routnig process. Apache passes that request string to the Moodle application who can then make its own service decision. NAT does not even touch that part of the header! It makes so much sense. I also enjoyed learning how to use the “apply as filter” tool in Wireshark to find the filter all of the GET requests in the session. Such a powerful tool!

You should be able to navigate to the moodles site via this URL:

http://73.174.113.166/moodle

You can login to the fake user and see my sample course:

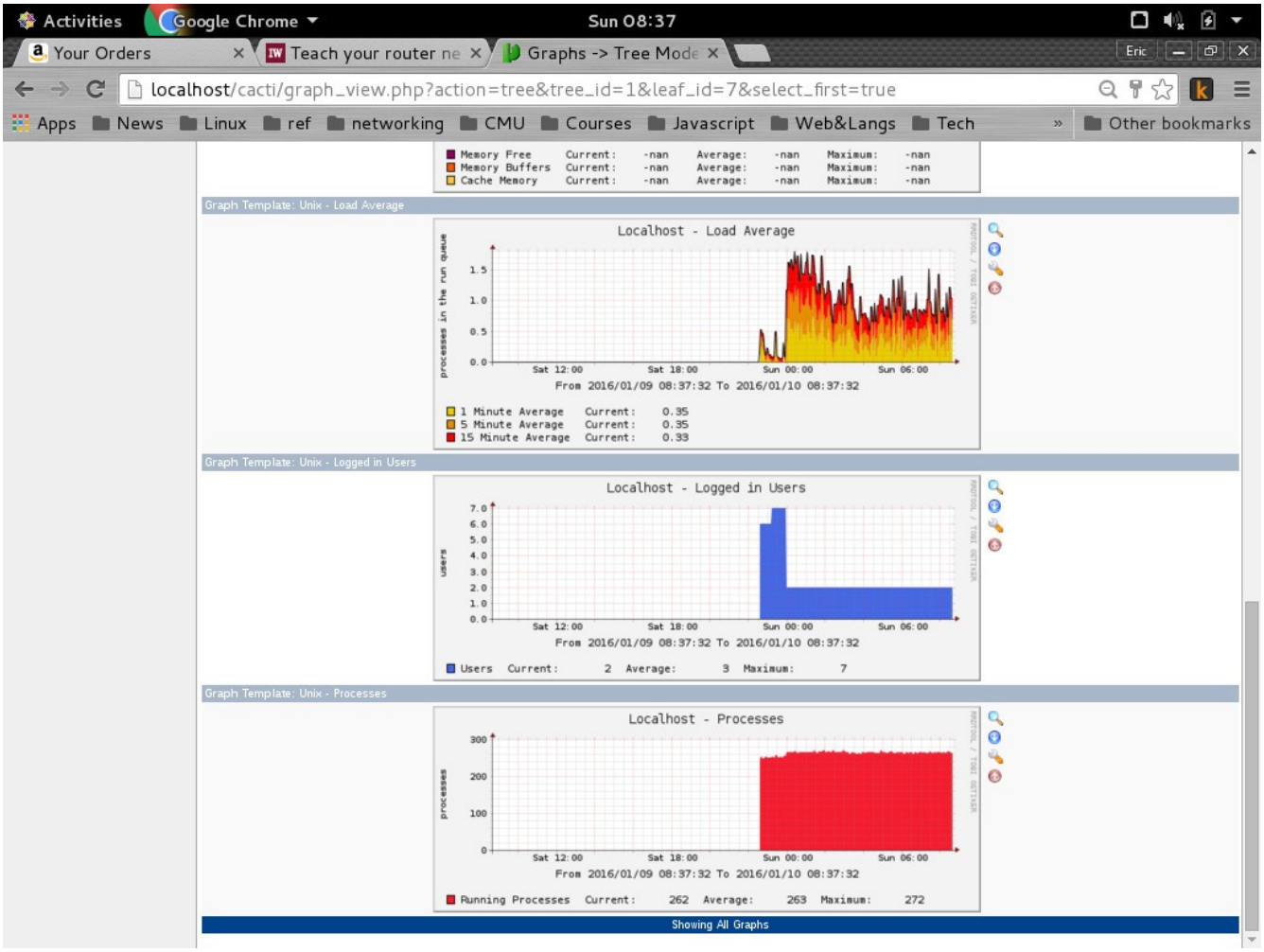
username: testuser

passowrd: E$n0wden

Of course this URL could change when the Xfinity lease is up, but it should work for the purposes of checking the system in the short term before the DDNS is up and running.

Setting up Cacti network monitoring & UPS daemon

Any good network and server is only as good as one can monitor it! I installed the popular Cacti monitoring software on the server and monitored content all night. Here's a screen shot of the server's memory usage, processes, and a few others.



This is the customized graph output of the Cacti network monitoring tool that uses SNMP packets sent from the SNMP daemon on the systems to be monitored. The system logs a status every five minutes and saves the data in a Round Robin Database that manages a static database file size even as monitoring continues.

I hit quite a few little detours while configuring cacti's SNMP monitoring service. The LAMP application installed successfully after I installed about a half dozen packages for the SNMP daemon and an Apache module. The Cacti setup screen doesn't default to working values, so I had to enable SNMP and choose the proper ports and timeouts, etc. Cacti also uses TCP ping packets in addition to SNMP packets to monitor host uptime, which seems like a solid component to have in case the SNMP daemon goes down for any reason. It wouldn't be helpful to have this show up as a complete server failure in the logs.

I also edited the SNMP server's configuration file to properly release server data to a 'community' group which meant that the packets going out every five minutes per the directive in the cron tool contained proper and accessible data. I didn't dig into exactly how all of that works, but I felt great that editing that /etc/snmp/snmpd.conf file seemed easy and natural when a few weeks ago, doing so seemed like quite a

hassle.

One can configure SNMP monitoring to be extremely versatile in what information it sends out with its packets. SNMP version 1—the version I'm using for Cacti—does not allow for authentication of the packets exchanged but later versions do so and the configuration file controls how all of that works. Much more sensitive information on the server can be sent out in this setup, even to hosts outside the LAN (not default). Out of the box, the SNMP daemon releases packets with very basic data only to localhost, which was just fine for my cacti application. Configuring release to hosts on the same LAN as the cacti-monitoried host is also straightforward and secure since it will all be behind the firewall. I'd like to work up to programming some basic applications that use this neat protocol for monitoring and signaling!

I found this tutorial on how to configure a router running Tomato firmware to release SNMP packets that Cacti can then monitor. It requires setting up another client/server pair using the common internet file system (CIFS) that allows a router with limited system capability to output useful data. This is part of LAN configuration phase 2.

https://systembash.com/how-to-install-snmp-on-tomato-router-firmware-and-graph-traffic-with-cacti/

As a final note, I should say that I was worried that my SNMP configuration in Cactus was incorrect on a confounding level because I could see that SNMP packets were being read correctly by the cacti application (as evident by a status dump in the localhost interface page) but even after an hour of recording data (i.e. 12 packets), the graphs displayed zero data. Frustrated that I couldn't find an easy fix, I went to bed only to wake up to see beautiful data! I learned that the round robin database tool (RDDTool http://oss.oetiker.ch/rrdtool/) that Cacti uses to store its data is designed for storing lots of time series data with decreasing density as time passes. This means that the database system needs a decent amount of data points before time-series averages and displays are possible. Whew! I would imagine that very view Linux bug fixes in my future will be as easy to fix as just *sleeping on it*.

Automatic power-down during power outages

My uninterrupted power supply device from Cyber Power sports a nifty USB interface which a power monitoring daemon process on any Linux machine can use to monitor the power state and take appropriate action in the event of an outage. The software was developed by Cyber Power and distributed in a convenient .rpm package which installed easily. Here is a screenshot of the daemon's status update. Note that it is ready and waiting for any change in status from the UPS device. I tested the system by running the daemon (it is set to load on system boot) and unpluggng the UPS from the wall to simulate an outage. The daemon initiated a system shutdown successfully 60 seconds into the outage.

This is a standard status output of the pwrstatd daemon accessed with the pwrstat tool. You can see it registers a normal UPS status and tells us when I did my test outage (at 8:31 pm evidently).

Reflections on Linux learning resources

My Linux explorations exist on three distinct 'modes' of operation:

1. Robust concept-driven learning: Comfortable learning and tinkering, stuck spots are eliminated with ease. I'm learning fundamentals that I can apply creatively as needed.
2. Health Problem solving: I'm focused on applying fundamentals but the bugs are difficult to understand and forum help isn't delivering the goods.
3. Grasping for miracle commands: “I don't care about what is going on, just work dammit!” This phrase muttered during mode 3 when stuck spots are too challenging to debug with current knowledge and often desperate forum searching ensues. Commands are blindly typed into command lines and silent prayers are said to unknown, unreliable gods.

The goal was, of course, to maximize productivity while still remaining in modes 1 and 2. During the beginning days of configuring software (Apache and Moodle), I was was pleasantly able to remain in modes 1 and 2 and avoid the desperation of mode 3. I would read entire chapters from the Sobell book before plunging into using a tool (say, `iptables` or `netstat`) to avoid the blind command line frenzy syndrome that is inherent in mode 3. The key to realizing this goal, I have found, is to not shortchange the fundamental learning of relevant tools. Man pages became my first line of learning rather than the forums. If one stays rooted in trying to understand what, exactly, a tool like like `systemctl` is doing, rather than just looking for a forum post that tells the reader about this or that 'magic option', the process of learning and using Linux becomes much less traumatic.

Forums are, of course, invaluable in debugging since there is so much variation in any given setup that a reference text or explanation tutorial can't realistically address every possible issue or conceptual topic that would allow one to debug issue X or Y successfully. A core practice, however, was for me to not just blindly follow forum advice and instead hold myself accountable for at least knowing exactly what each option is for each command I give. I would look up syntax issues online and then reference conceptual explanations in the Sobell text as I worked through this or that installation process.

While this probably quadrupled my overall time to task completion, the goal of this process was not task completion but robust tool learning. When I remained faithful to these principles, the process was overwhelming engaging and exciting. Only during moments of peril—inevitably late at night—would a forum search frenzy ensue which was almost always a sign that I needed to slow down and do more reading about the concepts that I clearly don't understand enough to debug on my own.

So often in our formal schooling process course assignments and their corresponding deadlines tend to push students like me into mode 3 thinking and trap them there for the duration of the course. This leads to a major deterioration of the quality of learning that results from the assignment and, when repeated, the entire course. My database class in the Fall, for example, was structured in such a way as to push me into frantic, surface-level groping inherent in mode 3 work many times. When the deadline was looming and the SQL is not running without error, I generally couldn't (or realistically did not have time to) go back and understand on a deeper level how client view variables are stored in the configuration databases, for example. If I had taken time for such detours, not only would I know a lot more about databases and be much more independent in my ability to use them effectively to solve new problems, but I would likely pick up many unexpected but useful ancillary knowledge elements.

My work on this project was notably different from the database coruse: the qualitative learning experience was much more engaging and the depth of my understanding that resulted was much more robust. This list captures the core attributes of the learning experience that enabled this kind of vibrant learning:

- Outcome parameters were specific enough to provide direction to the learning but didn't feel like shackles or do-or-die requirements. Renegotiating which app to install on the LAMP, for example, was helpful to avoid slides into Mode 3 learning. Once an outcome becomes inflexible and errors occur, the richness of the learning can often fizzle.
- I could comfortably spend time exploring related branches of knowledge that supported the core concept I happened to be learning. For example, I spent an hour or two learning how and toying with how boolean algebra is used to match host addresses with network addresses via the subnet mask. Had I been on a strict deadline to 'configure the subnet' I would have stopped once I found the proper `sudo route add -h …` command on a help forum. Now that I have taken such detours, I can speak and operate with much more robustness in the networking space compared to having just followed a tutorial.
- I think Linux is really, really cool and open source technology's community of committed, sharing folks is endlessly inspiring to me. I enjoyed reading the bios of the people (almost all men, sadly) who wrote the forum posts or the exceptional tutorials. As my introduction noted, this kind of technology tinkering has been of interest to me for a long time. Thus, a predisposition toward a subject matter is immensely important in driving the kind of joyful learning and its associated detours that made the project so rewarding.

Most useful online reference sites

These sites proved useful throughout the project and would be worth peeking at and perhaps bookmarking for future reference.

- http://www.yolinux.com/ Is a clearing house for Linux and networking links as well as a home to a number of very thorough configuration tutorials. This extensive networking tutorial was very handy to understand high level concepts and to access consolidated information on how Debian versus Fedora-based distros vary. http://www.yolinux.com/TUTORIALS/LinuxTutorialNetworking.html
- https://www.kernel.org/doc/man-pages/ The man pages, of course! Sometimes it was nice to view them online and click through the links to other pages. The linkedness of them is often lost when viewing in the terminal.
- http://www.gnu.org/manual/manual.en.html The GNU documentation would often provide useful detours into conceptual learning. Less troubleshooting, of course.
- Ask Ubuntu is obviously a clearinghouse for distro-specific issues. Since I was working on both Unbuntu 14.04 and CentOS at the same time, I found these forum posts very useful during every stage. http://askubuntu.com/questions/289559/how-can-i-create-a-windows-bootable-usb-stick-using-ubuntu was one helpful post on usb stick creating.
- Techmint has generally above average advice posts on may topics. http://www.tecmint.com/ifconfig-command-examples/
- Geek stuff posts are hit and miss, but there are lots of them: http://www.thegeekstuff.com/2012/04/route-examples/
- This super handy repo of 'cheat sheets' (formatted reference pages for printing) was great as I learned VI, Emacs, man, less, etc. http://www.nixtutor.com/linux/all-the-best-linux-cheat-sheets/
- IT Geared has a comprehensive networking reference that I didn't use much but found it linked often: http://www.itgeared.com/topic-8/basic-networking/
- This HOWTO like book is a very extensive reference for networking concepts: http://linux-ip.net/html/index.html
- This math professor's site has great SSH-specific and DD-WRT-focused tutorials that doesn't skimp on the concepts underlying the commands—golden! http://www.quarkphysics.ca/ssh/ssh_everything9.htm
- The DD-WRT wiki is a treasure trove of great firmware specific and general networking info: http://www.dd-wrt.com/wiki/index.php/Tutorials
- This IPTables online book by Oskar Andreasson seems to be the old but definitive guide on everything iptables. I want to read it through cover-to-cover! https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html
- Digital Ocean, the VPS company, runs a fantastic knowledge base that delivered again and again.

Their SELinux tutorial that walks through how SELinux affects the core LAMP applications was 5 hours of very good learning. I'm so impressed by the number of times a Digital Ocean page came up in my searches that I'm inclined to use their services when the time comes since they're contributing so much to the community. https://www.digitalocean.com/community/tutorials/an-introduction-to-selinux-on-centos-7-part-1-basic-concepts

GNUFree Documentation License

GNU Free Documentation License

Version 1.3, 3 November 2008

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent

modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a

translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.