x86 routers for Gigabit SQM with OpenWrt

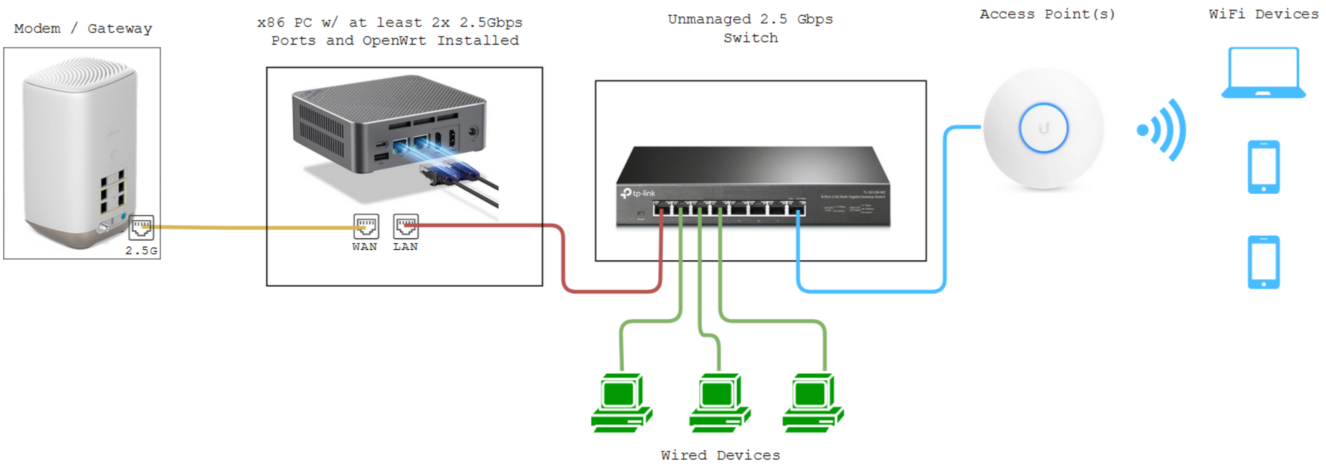
If you're here. It's because you want to get SQM for **1400Mbps+** and connections. The reason I'm saying 1400Mbps+ is because that is my current maximum provided by my ISP. The set up has potential for more than 1400Mbps. This is a write up based on an Intel N100 CPU mini PC I've tested.

If your internet connection is between **160-630 Mbps** it is more economical for you to get a NanoPi R4SE and follow the NanoPi guide instead. The NanoPi R6S is also cheaper and can do **1400Mbps+** as well, but keeps requiring constant CPU tweaks. For some reason it kept defaulting to the slower A55 cores instead of using the A76 cores every time I modify SQM settings.

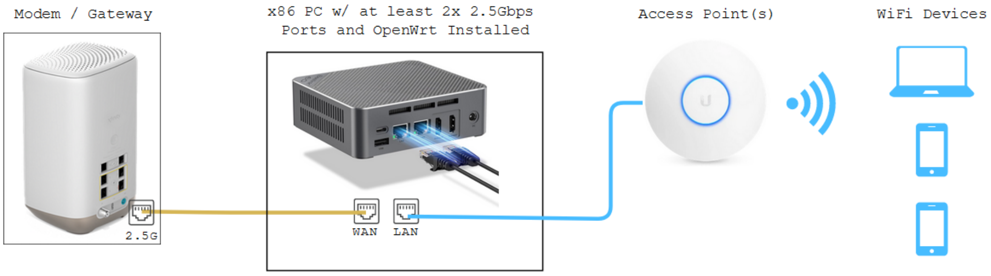
Now on to the physical setup.

The diagram below demonstrates how you would install any x86 Mini PC as a router in your network stack. Building your home network infrastructure like this, is more reliable and better than consumer routers which try to combine the modem, routing, and wireless all in one.

SQM Physical Setup For Wired + WiFi Devices (Beyond 1 Gbps)


Modem / Gateway
2.5G
x86 PC w/ at least 2x 2.5Gbps Ports and OpenWrt Installed
WAN
LAN
Unmanaged 2.5 Gbps Switch
Access Point(s)
WiFi Devices
Wired Devices

SQM Physical Setup For WiFi Only Devices (Beyond 1 Gbps)



Pictured Switch - TP-Link 2.5Gbps 8 Port Multi-gigabit switch **(Amazon Referral Link)** I earn a small commission on these links thank you!

Pictured Access Point - Ubiquiti Unifi 6 Pro (**Official Link)** Note: If Ubiquiti is out of stock you or if you don't like them, I heard that the TP-Link EAP670 **(Amazon Referral Link)** from their Omada lines work just as great. A reader of mine prefers Omada over the Ubiquiti stuff. I just have never tried it myself so I can't say much.

Pictured OpenWrt Device - BeeLink EQ12 Intel N100 Mini-PC **(Amazon Referral Link)**. If you're cheap and don't mind waiting you get an N100 x86 miniPC from Aliexpress. I had good experience with the one on the WooYi Storefront.

If you want a beefier PC that's probably overkill the AceMagician **(Amazon Referral Link)** with the AMD Ryzen 5800U or the Beelink EQR6 **(Amazon Referral Link)** would work. The over point is to make sure you get an x86 PC with at least 2, 1Gbps RJ-45 Ethernet ports.

What is bufferbloat and why do we not want it? It is lag or ping spikes in video games or zoom calls that is caused when you or someone else uses up all your bandwidth. It could be torrenting, 4k streaming, bulk downloads, or even a speedtest. SQM algorithms (fq_codel or cake) which are available on OpenWrt, can completely mitigate these pings and ensures low latency even under full load. Overall, you do sacrifice a little max speed 5-10% for guaranteed low latencies.

Hardware Requirements

The minimum requirement for choosing one of these x86 based PCs is as follows

1. Make sure it has at least **2x 1 Gbps Ethernet ports or better**. Intel Ethernet ports are generally preferred over Realtek ports but Realtek ports will do. You might want to consider 2.5 Gbps if your ISP (Internet Service Provider) has the capability.

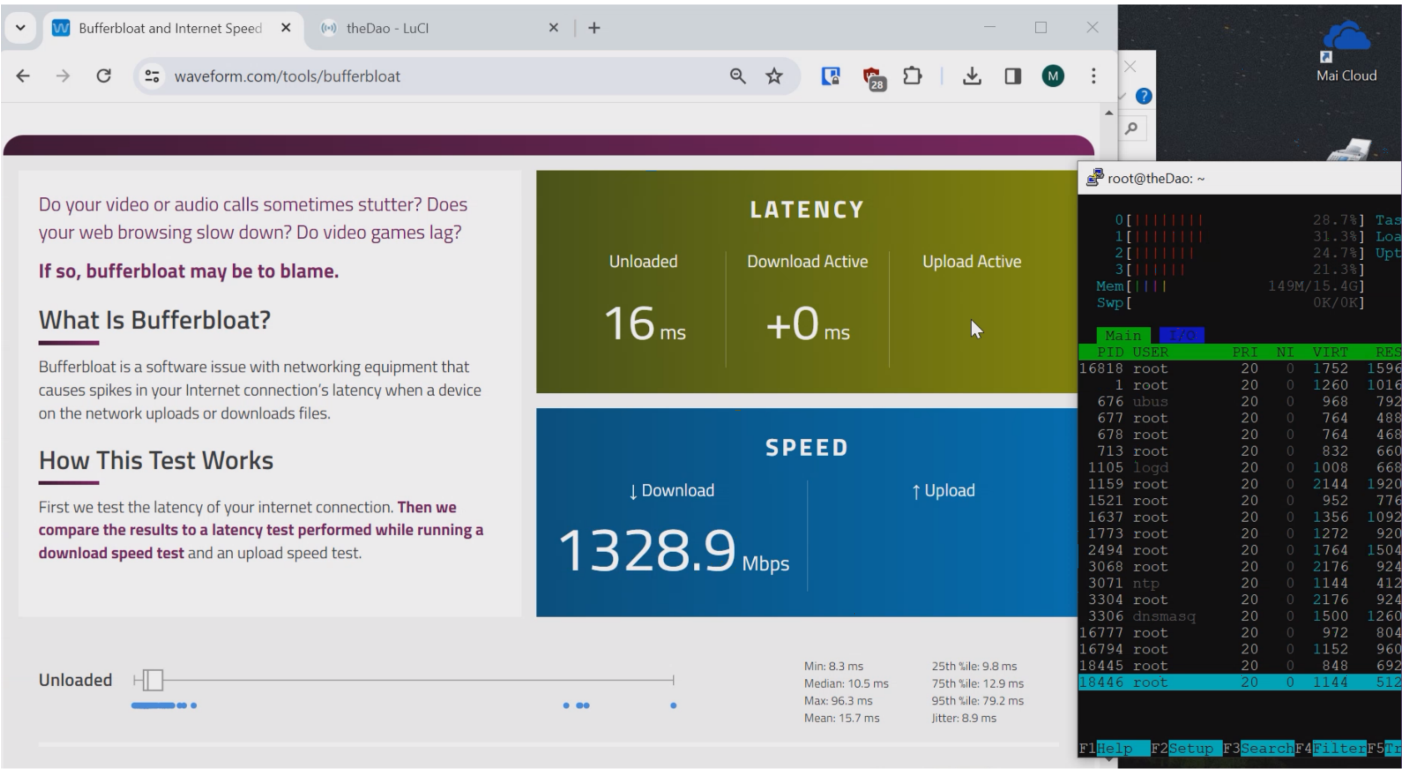
2. You also want to make sure it has a CPU Mark of 5000 or more. And a strong single core performance of 1200+. You can check here: https://www.cpubenchmark.net/cpu_list.php

3. Preferably it would be a low power device that uses < 25 Watts.

Intel Alder Lake N100 12th Generation Based Mini PCs

Performance of N100

On my 1400 Mbps connection I was able to run cake SQM at 1350Mbps with up to 40% CPU Usage on a single CPU core so there's a lot of room for more Mbps!



Do your video or audio calls sometimes stutter? Does your web browsing slow down? Do video games lag?

If so, bufferbloat may be to blame.

What Is Bufferbloat?

Bufferbloat is a software issue with networking equipment that causes spikes in your Internet connection's latency when a device on the network uploads or downloads files.

How This Test Works

First we test the latency of your internet connection. **Then we compare the results to a latency test performed while running a download speed test** and an upload speed test.

Min: 8.3 ms
Median: 10.5 ms
Max: 96.3 ms
Mean: 15.7 ms

25th %ile: 9.8 ms
75th %ile: 12.9 ms
95th %ile: 79.2 ms
Jitter: 8.9 ms

root@theDao: ~

```
    0[|||||||||                28.7%] Tas
    1[|||||||||                31.3%] Loa
    2[||||||||                 24.7%] Upt
    3[|||||||                  21.3%]
  Mem[||||                 149M/15.4G]
  Swp[                          0K/0K]

   Main   I/O
  PID USER       PRI  NI  VIRT   RES
16818 root        20   0  1752  1596
    1 root        20   0  1260  1016
  676 ubus        20   0   968   792
  677 root        20   0   764   488
  678 root        20   0   764   468
  713 root        20   0   832   660
 1105 logd        20   0  1008   668
 1159 root        20   0  2144  1920
 1521 root        20   0   952   776
 1637 root        20   0  1356  1092
 1773 root        20   0  1272   920
 2494 root        20   0  1764  1504
 3068 root        20   0  2176   924
 3071 ntp         20   0  1144   412
 3304 root        20   0  2176   924
 3306 dnsmasq     20   0  1500  1260
16777 root        20   0   972   804
16794 root        20   0  1152   960
18445 root        20   0   848   692
18446 root        20   0  1144   512

F1Help  F2Setup F3SearchF4FilterF5Tr
```

Installing OpenWrt

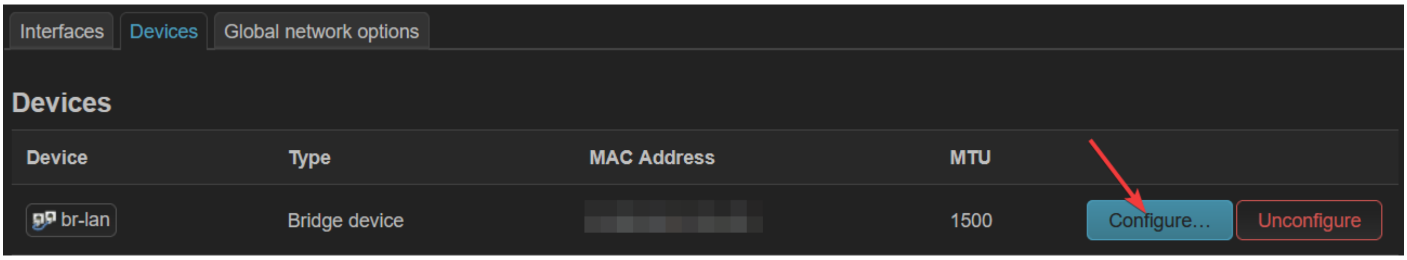
As for installation of software. OpenWrt has an official written guide here:

https://openwrt.org/docs/guide-user/installation/openwrt_x86

If you're lost on what to do no worries I can give some pointers. I typically choose the generic-squashfs-combined-efi.img.gz image

1. I normally would take out the nVME SSD of my miniPC to an external enclosure.
2. Plug it into my primary PC.
3. Then flash the official image (using win32diskimager or balenaetcher for windows) it to my Router PC's nVME SSD inside the enclosure.
4. Then I would take the nVME back out of the enclosure and install it back in the miniPC.
5. That's pretty much it! OpenWrt should boot from there. The default ports for WAN and LAN may incorrect such as ETH0 being LAN and ETH1 being WAN but we can fix that.

Optional (Swapping ETH0 to WAN and ETH1 to LAN): This is done in Network > Interfaces -> Devices Tab -> Then hitting configure under the device called "br-lan"/



Devices

Device	Type	MAC Address	MTU	
br-lan	Bridge device	[redacted]	1500	Configure… Unconfigure

Attach eth1 as the bridge port. You can also attach eth2 eth3 if you have more ports, just make sure you detach eth0.

Then go back to Network > Interfaces. Edit "wan" and "wan6" so that they use eth0 instead of eth1.

Then save and apply. Plug your machine to ETH1 (or ETH2 ETH3 if you have those.... and Modem into ETH0.

Optional (Expand rootfs to use all SSD size) : Now the problem with all the openWrt images is that the root partition is small by defualt. The easiest thing you can do to expand this size is to boot a live ubuntu usb disk. Run gparted and expand the rootfs partition so you can utilize the full size of your SSD.

After OpenWrt is setup and running you just need to enable SQM via the official openWrt guide or my guide.

Either way feel free to improve it further with the advanced cake config section of this page

Gotchas

If you're not getting your beyond Gigabit speeds make sure you're using CAT6 or higher Ethernet cables and make sure the devices like the laptop you're using also have 2.5 Gbps ports.

In the diagram, I made sure that the example router, switch, and access point all had 2.5 Gbps ports!

What Access Point to Get?

I keep hearing raving reviews about the Ubiquiti APs and use one myself. I have extremely stable WiFi with these and never have to reboot them. Ubiquiti also advertises up to 200 concurrent users as well! If you have a recommendation better than these I'd like to know.

Ubiquiti Unifi 6 Pro (**Official Link)**

If you plan on only having one Ubiquti AP I recommend installing via the phone so you don't have to bother with more complicated things like AP Controllers.

If you're on a budget and can't buy a dedicated AP. You **can try turning your old router** into an **access point** by putting it into AP mode instead of routing mode. This is important because you should be letting the OpenWrt device do the routing to prevent bufferbloat not your old router.

Another option you could try that I've heard are good are the TP-Link EAP670 **(Amazon Referral Link)**. I have no real world experience with these as I don't own any, but I heard they are solid products in the /r/homenetworking community.

Facts about WiFi

If you need more coverage you should get more APs not one single AP with a bunch of antennas, because those are marketing gimmicks.

WiFi has limited range due to the physics of their frequency bands.

5Ghz can handle more bandwidth, but will usually be about half the range of 2.4Ghz.

Advanced Cake Configuration

This section is for my own reference and these were recommended by the official docs:

https://openwrt.org/docs/guide-user/network/traffic-shaping/sqm-details#sqmqueue_discipline_tab

It's not necessary to do this but if you want even further ping stability under load it might be worthwhile!

Under the Queue Discipline tab of SQM.
Enable the checkmark for advanced configuration and save& apply.

This turns on squash_dscp, squash_ingress, ECN on ingress and NOECN on egress. Leave them as defaults as they are good the way they are. (If you have symmetrical fiber then ECN can be enabled on egress.

Next checkmark and enable "Dangerous Configuration" which is below the "Advanced Configuration" section. We are going to disable triple-isolate and enable per host isolation... Here's a short explanation.

To quote the docs, by default, cake will use triple-isolate: “which will first make sure that no internal or internal host will hog too much bandwidth and then will still guarantee for fairness for each host. In that mode, Cake mostly does the right thing. It would ensure that no single stream and no single host could hog all the capacity of the WAN link. However, it can’t prevent a BitTorrent client – with multiple connections – from monopolizing most of the capacity.” You can enable per host isolation, which will identify all source/destination information.

To enable that, Add the following to the “Advanced option strings” (in the Interfaces → SQM-QoS page; Queue Discipline tab, look for the Dangerous Configuration options):

For queueing disciplines handling incoming packets from the internet (internet-ingress): nat dual-dsthost ingress

For queueing disciplines handling outgoing packets to the internet (internet-egress): nat dual-srchost

For me that means Qdisc options (ingress) I wrote in "nat dual-dsthost ingress" while for Qdisc options (egress) I wrote in "nat dual-srchost"

Optional: Docker on x86

This section is for those of you who run docker containers.

For x86 machines you'll need to run the following commands in order for docker-compose to work and wg-easy docker container to work. For some reason the default opkg install docker-compose does not seem to work, so the solution is below.

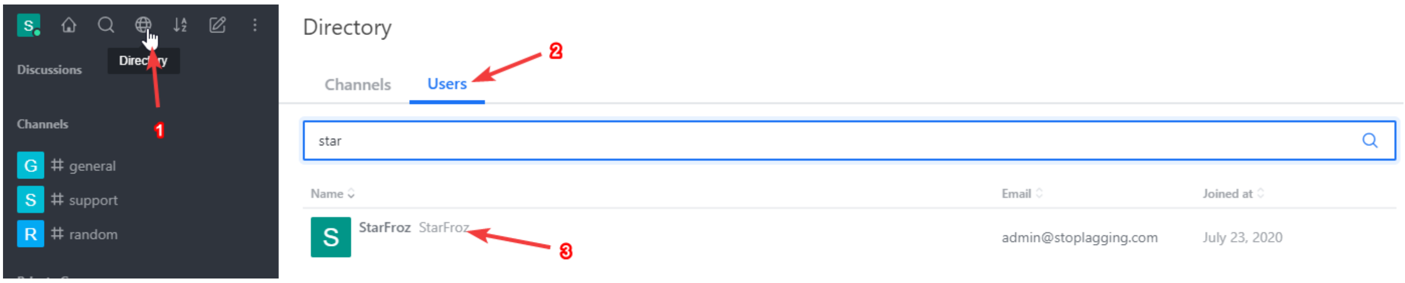
```
The first command line:
opkg remove docker luci-app-dockerman docker-compose dockerd --autoremove --force-remove

Then command line:
opkg install docker luci-app-dockerman docker-compose dockerd --force-maintainer

opkg install luci-proto-wireguard
opkg install wireguard-tools
```

Contact

If you need help or consultation please join my rocket.chat server at https://chat.stoplagging.com/invite/zaMu6X you can message me @Starfroz by looking me up under the globe icon after registering and logging in.



star

Name	Email	Joined at
StarFroz StarFroz	admin@stoplagging.com	July 23, 2020

Revision #48
Created 7 January 2024 00:28:34 by Admin
Updated 27 September 2024 18:54:29 by Admin