Tackling the Internet Edge Problems

Jim Gettys – jg@freedesktop.org
Dave Taht – dave.taht@gmail.com
Version 0.0, July 21, 2011

Solving the Home Router Disaster

Home routers are a mostly unrecognized major problem for the Internet. Home routers have serious problems with bufferbloat, IPv6, and security.

Commercial firmware for most of the market is based upon Linux kernels and other open source projects, but the firmware is generally unmaintained after release; worse yet, the code base they use is over five years old! This dysfunctional behavior is all-too-common in the embedded Linux market, which both of us helped spark over a decade ago in our respective careers, but failed to prevent despite our attempts.

The incumbent vendor's current code bases are therefore useless as a platform for research and development as any patches for problems are impossible to integrate easily into the upstream projects. Competitive pressures among vendors make it hard for major vendors to want to contribute to solving the problems if their competition will benefit.

There is an alternative bleeding edge and open code base available in the OpenWrt project.

OpenWrt is already good enough that older versions of OpenWrt are shipped in some of the smaller commercial home routers.

Home routers using firmware from OpenWrt have matured to where modest amounts of labor can succeed in a such a technology demonstration, providing the “proof of principle” to encourage the embedded network industry to move quickly to address these issues.

CeroWrt is an OpenWrt router platform for use by individuals, researchers, and students interested in advancing the state of the art of the Internet. Specifically, it is aimed at investigating the problems of latency under load, bufferbloat wireless-n, QoS, security and the effects of various TCP algorithms on shared networks.

CeroWrt is motivated by the following observations:

1. A standardized test platform is needed for testing AQM algorithms. Duplicatable results are needed from diverse wireless environments, which presents a much harder problem than Internet core routers. AQM algorithms which work in this environment may or may not be applicable to broadband devices and core routers. To solve both bufferbloat and IPv6 problems, we need an inexpensive test platform on which we can perform tests in diverse environments.
2. Bufferbloat, IPv6 and security problems are most severe in home routers.
3. IPv6 is not deployable without name services: the addresses are effectively impossible to type by mere mortals and management of a home network is untenable
4. The home router market is typical of the embedded Linux market: it is dysfunctional, often shipping firmware five or more years obsolete at FRS. While we can fix upstream Linux and other projects relatively quickly, the dysfunctional embedded market leaves home routers crippled indefinitely unless the market can be shifted

5. A few of the smaller home network vendors already ship OpenWRT based products.
6. A existence proof of a home router working well will enable the ISP's to exert full pressure on the home router market; but without an existence proof, it is unlikely the market place will quickly provide one.
7. Security of home network devices is anywhere from poor to horrific; firmware is not updated once it is stable in a device, and then moulders until that hardware is retired, inviting attacks on the routers; we await the day the first major disaster occurs, but it is only a matter of time.
8. End users do not have any assurance of DNS lookup integrity: DNSSEC needs deployment all the way to the edge of the network

In essence, this proposal is to augment the OpenWRT project to tackle the modern home router disaster, to take OpenWrt from a niche for wireless enthusiasts to a real proof of principle meeting the needs of researchers, and then the mass market, enabling rapid adoption of the needed solutions for IPv6, bufferbloat and security in the home router market.

All work will be open source and be in and with the upstream projects to enable the existing home router industry to pick up improvements as rapidly as possible, whether or not they directly use the OpenWRT/CeroWrt code base. Fixes will be routinely pushed “upstream” to the key open source projects from which OpenWrt is derived: e.g. kernel.org, ISC bind, busybox, etc. Thus, even if vendors never directly use the results of CeroWrt, at least the results may eventually trickle into their firmware when they belatedly upgrade to new versions of those packages.

Features

CeroWrt is a OpenWrt build specifically tailored to meet the observations above, initially targeted for the Netgear WNDR3700v2, a modern dual radio 802.11n home router for which there is 100% open source Linux support, enabling debugging of all parts of the system. This router also has sufficient flash memory to lift some of the size constraints that have made some potential solutions difficult, such as using ISC's Bind, which already supports DNSsec, which dnsmasq does not..These routers are widely available for about $130, putting them in reach of almost anyone, and we can expect the price point of such routers to drop. To be confident in solutions for bufferbloat and IPv6 both, we must build a large community of contributors, developers, researchers and users.

CeroWrt currently includes:

- Current Linux kernel (2.6.39), with additional TCP (westwood+, reno, veno, cubic, bic) and AQM algorithms (SFB, DRR) that are not normally enabled.
- Preliminary debloating of excessive buffering in the wired and wireless stacks
- A number of recent ECN fixes that have been pushed upstream into the latest Linux releases for testing. ECN is enabled in these routers.
- ISC Bind enables “plug and play” IPv6 naming and publication of names into the global DNS; plugging a named system into a home network should enable it's named access from anywhere, without any manual configuration or intervention from the user.
- ISC BIND to enable end-to-end name service integrity using DNSsec

- Extensive network test tools for debugging/instrumentation/benchmarking of home network behavior
- IPv6 support: 6to4 and 6in4 is available and 6to4 is enabled by default if a globally routable IP address is available.
- Web server (lighttpd) and polipo caching web proxy.
- Luci web interface

While the CeroWrt distribution has (just) been integrated to the point of demonstration, the work has only just begun. The integration was relatively easy: the reduction to near production use and widespread deployment and testing will be hard. Funding is needed to move the project along in the following areas:

- QA and systematic testing of the distribution
- Development of scripts and test tools for evaluating AQM algorithms such as SFB and RED Light has yet to begin; these will require extensive testing and validation before deployment. This testing must be made in real diverse environments. Asking everyone to “roll their own” router distribution is more work than most people can do; installing a prebuilt image onto standard hardware enables much wider testing.
- 3g and 4g support and testing via USB dongles; the problems here are both similar and different than 802.11, due to the technology similarities and differences.
- Appropriate classification and shaper scripts; Diffserv support in wireless and in the router to control marking of packets and interact properly with 802.11 QOS features
- DNSSEC requires having valid time for it to function: solving this “chicken and egg” problem is required to have DNSSEC enabled at all times.
- IPsec based VPN support, with support for two factor authentication
- The experiences of since January show that any code in the networking stack that has never been used or deployed has bugs. We've help fix multiple bugs in ECN present for a decade. There will be more...
- Wireless drivers themselves are problematic, attempting much too aggressive retransmission in the face of packet loss, increasing effective bloat
- Performance and feature work in general: any “proof of concept” router needs competitive performance and essential features found in existing commercial firmware to be seen as credible to exert market pressure. There are known and unknown problems in this area to be overcome.
- User interface work for DNS & DHCP, and for providing a “simple” flag in the Luci interface in general, to enable a larger user base and project. Most users should not be presented with all the ways you can tweak a home router, which is needed to meet the overall goal of the OpenWRT project, which includes enthusiasts in community networking with additional requirements that are beyond most users
- OpenWrt is already a package based system which allows for updating of devices in the field: however, its package system (opkg) does not as yet have support for digital signatures on those packages

- Over time, additional hardware devices will need support both for the base system IPv6 and bufferbloat work, but also to enable supporting a larger community of users

Oversight and Organization

The stakeholders include:

- The OpenWrt project
- ISOC?
- Funders (Google? Major ISP's?)
- Researchers, who need a platform for development in these areas
- Router vendors, if/when they choose to become involved
- The community of internet users

An advisory committee is needed to express the needs of these communities to the project.

The budget below is focused on the particular goals expressed above, and to provide the development infrastructure and framework to enable funding to be used wisely and rapidly to address the issues at hand. As in any open source project, constituencies desiring particular features beyond the immediate bufferbloat, security and IPv6 goals should provide sweat equity in the OpenWRT/CeroWRT project more than regarding this project as a consortium that is directed by its members.

Avoiding mission creep is necessary.

In addition to developing a platform capable of repeatable results there are multiple sub-projects that can spin independently, the development of a testbed that can be built at scale, improvements to existing test tools, and the engagement with researchers and students in academia.

The Testbed Problem

Van Jacobson notes that it's a huge gulf between minimum publishable unit and mass deployment. How do we get rigorous testing of AQM/ECN etc? With 802.11 driver and technology problems mixed in, this is non-trivial.

Open source developers can't easily test at scale with many devices and multiple routers. Doing preperatory work at for testing at scale needs to get moving as it takes time to setup. We're nearly ready to start testing...

Emulab may be an answer here: but they'll need to update their hardware and probably need some funding. If not Emulab, UNH IOL?

How do we get researcher's engaged on the issue? Who to talk to? I knew Jay Lepreau, but don't know any of the others? How to find someone to lead the testing charge? Funding for upgrade and someone to lead the testing effort?

How to spark such research? What meetings should CeroWrt get pitched to?

How to ensure the Linux community's involvement?

The Test Tools Problem

Better bufferbloat/ECN detection/debugging tools are needed.In addition to tools that are effectively orphaned, such as pathchar/pchar, and there are many, such as netperf, that can be improved.

Government funding currently has no model to follow through on research tools that are successful (examples: pathchar, or netalyzr). Can this be fixed? Or is it easiest to seek other funding? M-labs? The tools to debug the network are not seeing proper development and maintenance (with a few notable exceptions, such as wireshark).

Who is the right person to lead a focussed charge here?

Dataset Problem

How do we collect and thoroughly analyze what we are seeing? How do we get engagement from the research community? How do we bridge the gap between that community and the open source community?

End-to-end systems testing and debugging is hard, particularly if you don't happen to stumble into a simple test case as Jim Gettys did for bufferbloat. I knew just enough, and happened to know the right people, to send the weird traces I took to people who could confirm the problem. And, with Netalyzr, we knew how widespread the problem was.

Bufferbloat has been a significant problem for **at least** seven years, but no action has resulted due to insufficient analysis, and similar problems encountered again and again since the development of the ARPAnet. When weirdness is detected, how do the experts get called in?

Budget

A project lead for CeroWrt - Dave Täht

AQM testing lead - ???

AQM research funding

Probably a few others full time.

Budget for the related organizations.

Much of the rest of the work is naturally able to be done via contract. Various other roles such Sysadmins, build engineers, packagers, test designers, QA, device driver developers, web developers, etc, can be hired or contracted on an as needed basis, and pulled from the 260 volunteers currently participating in the ongoing Bufferbloat effort.

There will be travel and lab expenses as well.