FreeBSD 2026: Development Proposal

Dr. Bruce Simpson

FreeBSD Developer Summit, Ottawa, June 2026

I. Shut up and code is no longer enough

I am Bruce Simpson, Ph.D, a domain expert in IP networking and telecommunications; to some, “just another old school networking hacker”. I have done a significant amount of paid and pro-bono open source software development over the past ~25 years, mostly within FreeBSD itself. I am currently returning to FreeBSD after a 7 year sabbatical due to ill health and multiple bereavements. However, my track record on delivering RFC-compliant networking software is clearly established.

My consulting entity, Routeplex Limited, has full professional indemnity cover, and is currently self seed funded. I intend to seek micro-capital funding from Highlands & Islands Enterprise, Scottish Enterprise, British Business Bank through its regional partners, and the EU Horizon Europe scheme in the first instance, with funding calls pending over April 2026.

Routeplex Limited’s activities are semi-commercial; I am looking to commercialise some of the research I conducted at the University of St Andrews over 2011-2016, although I foresee much collaboration with the FreeBSD Foundation and other stakeholders (e.g. HPE Juniper, Deciso BV, and Netgate in the first instance). Whilst the scope of this overall work plan is ambitious, tasks must be prioritized according to available resources. Opportunities exist to delegate the work where that makes sense.

II. eBPF in FreeBSD

The Enhanced Berkeley Packet Filter (eBPF) is a refinement of the original BPF which originated in 4.4BSD UNIX; it, in turn, originated within the Linux community. eBPF is somewhat different from BPF in its scope & intent, although the basics are similar; it extends the BPF virtual machine to full 64-bit width. A requirement for eBPF in FreeBSD was identified as a key strategic infrastructural step for Routeplex Limited’s semi-commercial aims. The FreeBSD Foundation is a definite stakeholder in this effort; a detailed work plan must be prepared and submitted.

Dr. Hiroki Sato must also be actively engaged in this effort, as it was one of his students who developed an Apache 2.0 licensed kernel implementation of the eBPF virtual machine: generic-ebpf. I have extensive upfront research notes on eBPF system function and potential FreeBSD adoption. Please contact me out-of-band if so interested. It is also known that there is a compiler for the P4 packet processing language which targets the eBPF virtual machine: p4ebpf, and this is of semi-commercial interest, also.

Dr. Bruce Simpson, Director, Routeplex Limited, Argyll & Bute, Scotland, UK. E-mail: bms@FreeBSD.org and/or bms@routeplex.co.uk

Linux-oriented documentation increasingly refers to eBPF as just "BPF", which can cause much confusion. From the documentation for Cilium, a major eBPF consumer: “Nowadays, the Linux kernel runs eBPF only and loaded cBPF bytecode is transparently translated into an eBPF representation in the kernel before program execution.” So, cBPF is still supported in Linux, and is used by e.g. libpcap through established methods. The Linux community now promotes eBPF as an alternative to DTrace. This can be seen in the syntax for tools like “bpftrace”. It has been suggested that eBPF now outperforms DTrace at the same system instrumentation tasks, as per Mateusz Piotrowski’s presentation at BSDCan 2023 which provides a good contrast analysis.

Adopting eBPF is especially compelling given some of the new features that Cloudflare have contributed to Linux itself, e.g. modifying the IP transport protocol 4-tuple lookup to accomodate arbitrary IP network prefix wildcards. These approaches are worth looking at for many other applications. However, they did not extend this to SCTP or other transports. It is also worth looking at this approach, as some of the solution space overlaps with the issues the Identifier-Locator Networking Protocol (ILNP) has with state implosion for 1:M UDP, anycast, and multicast.

The Packet Construction Set (pcs), described further on herein, has adopted the cBPF naming convention, following contemporary libpcap itself. As far as the docs for Cilium described the basic opcodes, they seemed to directly overlap with those I had defined in pcs, which I partially factored out from Dug Song’s original PCAP binding, pypcap, in the beginning.

III. TCP Enhancements

A. TCP-AO: TCP Authentication Options

The TCP Authentication Option is an extension to the protocol defined in RFC 5925, with its requisite cryptography algorithms defined in RFC 5926. As the Internet has grown, various Internet Exchange Points (IXP) have begun to accomodate or require the use of TCP-AO to offer additional security above and beyond that offered by RFC 2385 TCP-MD5. Over the course of the 2010s, Andre Oppermann was funded to deliver the feature, but for whatever reason, it did not land. Both Rui Paulo and Bjoern Zeeb had expressed an interest in finishing the work, but again, did not get around to it.

Andre’s code comments suggest an intent to refactor TCP-MD5 in terms of the TCP-AO implementation. This would be desirable as the use of the IPSEC Security Association Database (SADB) for TCP-MD5 was not appropriate, and reflected implementation expedience at the time. By contrast, the RFC 5925 implementations for other operating systems use socket-wide (and not system-wide) keying databases.

As of writing, John-Mark Gurney's employer HPE Juniper have expressed informal interest in the work. Routeplex Limited has solicited them with a semi-formal Letter of Intent (LoI) to conduct this and potentially other FreeBSD work on a semi-commercial basis. The introduction of these features mandates addition to the opencrypto framework (as instigated by Sam Leffler in 2003), and this presents refactoring opportunities for carp(4) and other kernel consumers where HMAC-SHA2-256 is being hand-rolled, as well as future consumers (e.g IPv6 Segment Routing (SRv6), and distributed router extensions to RFC 6740+ ILNP).

B. Delayed ACK Profiling for BBLog

The Delayed ACK/Nagle Algorithm implosion problem is a well known issue with the TCP protocol when employed for interactive session use and/or Remote Procedure Call (RPC) applications. In the first instance, I do not seek to attempt to solve this problem; it is uncertain whether it can ever truly be solved without employing additional workarounds such as Nagle himself has already described. Instead, I propose to extend the TCP Black Box Recording feature added by Randall Stewart and Michael Tuexen to provide profiling for the situations in which it can occur.

I have however conducted initial research into how the state-of-art mitigation mechanisms might be implemented within FreeBSD. Please contact me out-of-band for more information. In the meantime, an excellent summary of the problem space can be found at Dr. Stuart Cheshire's personal web site; as of writing, he remains scientist-in-residence at Apple.

Nagle's Algorithm exists to prevent the emission of "tinygrams". Often, software consumers of such will explicitly disable the use of the algorithm by immediately setting the TCP_NODELAY option on a freshly accepted or connected TCP socket. John Nagle himself opined on Y Combinator Hacker News: "A delayed ACK should be thought of as a bet on the behavior of the listening application. If the listening application usually responds fast, within the ACK delay interval, the delayed ACK is coalesced into the reply and you save a packet. If the listening application does not respond immediately, a delayed ACK has to actually be sent, and nothing was gained by delaying it."

He continued: "Delayed ACKs are a win only in certain circumstances - mostly character echo for Telnet. (When Berkeley installed delayed ACKs, they were doing a lot of Telnet from terminal concentrators in student terminal rooms to host VAX machines doing the work. For that particular situation, it made sense.) The delayed ACK timer is scaled to expected human response time. A delayed ACK is a bet that the other end will reply to what you just sent almost immediately. Except for some RPC protocols, this is unlikely. So the ACK delay mechanism loses the bet, over and over, delaying the ACK, waiting for a packet on which the ACK can be piggybacked, not getting it, and then sending the ACK, delayed. There's nothing in TCP to automatically turn this off. However, Linux ... now have a TCP_QUICKACK socket option. Turn that on unless you have a very unusual application."

It is known that the Apple macOS xnu kernel implementation of TCP, partially derived from FreeBSD's implementation, has such Delayed ACK profiling, although it is not as readily available to systems administrators and users as I propose. Upon source code inspection, I observed no delayed ACK counters or profiling in Linux, and there appeared to be no facility similar to BBlog, unless eBPF were employed specifically to implement this feature. Apple's implementation appears to have closely followed Nagle's advice, almost to the letter: "It would be useful for TCP implementations to tally, for each socket, the number of delayed ACKs actually sent vs. the number coalesced. If many delayed ACKs are being sent, ACK delay should be turned off, rather than repeating a losing bet."

Finally, Nagle had this to say about the history of the D-ACK implosion problem: "The real problem is not tinygram prevention. It's ACK delays, and that stupid fixed timer. They both went into TCP around the same time, but independently. I did tinygram prevention (the Nagle algorithm) and Berkeley did delayed ACKs, both in the early 1980s. The combination of the two is awful. Unfortunately by the time I found about delayed ACKs, I had changed jobs, was out of networking, and doing a product for Autodesk on non-networked PCs."

C. Minshall's extension to Nagle's Algorithm

Both Cheshire and Minshall caution against disabling Nagle's Algorithm. Deriving a clean-room implementation of Minshall's modification specifically for FreeBSD's default TCP functional block ("TCP stack") is a proposed engineering task. This effort follows on naturally from the proposed work to implement Delayed ACK profiling in FreeBSD. The Apple macOS xnu kernel already implements the Minshall modification, as does Linux. The xnu implementation cannot be taken directly due to its incompatible APSL licensing and subsequent code divergence.

To understand this in depth, a more modern description of Nagle's Algorithm and its application in TCP today can be found in Sec. 3.7.4 of RFC 9293, the updated TCP standard. This mentions the delayed ACK implosion problem, and refers to a possible adjustment to the algorithm documented in expired I-D draft-minshall-nagle-01, which providers a fuller exposition of the problem space. Historically, the eXtensible Open Router Project's (XORP) implementation of RPC, XRL, disabled it immediately upon TCP socket creation, and coalesced large writes to avoid implosion with delayed ACKs in its AsyncFileWriter C++ implementation. More recent RPC frameworks appear to follow the same pattern.

IV. Ethernet Enhancements

A. IEEE 802.1ag: Connectivity Fault Management (CFM)

Currently, IEEE 801.ag CFM is not widely implemented in open-source operating systems. To some, this feature set is popularly known as "Layer 2 Ping" and "Layer 2 Traceroute", and it has specific utility in networks which are routed using the IS-IS protocol, where IP addresses of any kind are essentially optional. It is a necessary prequisite for "Carrier Ethernet": features used to manage the diagnostic state and

ongoing, in-line performance measurement of Ethernet in the First Mile (EFM), corresponding to standards and work practices adopted and advocated by Mplify (formerly, the Metro Ethernet Forum).

In the first instance, I propose to finish the work of adopting and maintaining the BSD-licensed, user-space implementation, dot1ag-utils, originating from sara.nl, followed by the constrained scope of a FreeBSD kernel-based "down" Maintenance End Point (MEP); the terminology of "down" indicates that the MEP is bound to a particular physical or logical link, rather than being node-wide as is the case with an "up" MEP. The use of a kernel-space responder is preferable to achieve timely, low-latency system response, and to co-exist with other Layer 2 protocols, such as IEEE 802.1AX(tm) Link Aggregation Control Protocol (LACP). Therefore, some refactoring of existing kernel code in if_bridge(4) would be required. This is also a prerequisite for the G.8032 work described further herein. External stakeholders such as Deciso BV and Netgate are strongly encouraged to participate in this work.

B. ITU-T G.8032: Ethernet Ring Protection System (ERPS)

The G.8032 specification provides an alternative approach to Ethernet Layer 2 loop detection, prevention, and failover, in the absence of, or co-existing with, LACP. It is arguably of more interest to physical, rather than logical, Ethernet switches. Given the limited time available, I would be looking to delegate this task to a fellow FreeBSD developer if so interested.

This work was originally proposed by Ellis Melman, formerly of Red Raw Internet (a UK WISP and former client of mine preceding Routeplex's existence), as an extension to FreeBSD to implement the RFC 3619 Extreme Active Protection System (EAPS), which preceded the ITU-T G.8032 specification (and was wholly specific to Extreme Networks switching products, with few exceptions). This was specifically intended to enable pfSense-based provider edge routers at serviced office buildings throughout London as they existed at that time, to interconnect to a city-wide Ethernet ring over optical fibre without requiring interposed Extreme switches.

Normally, the IEEE 802.1Q-2022 Sec.8 Spanning Tree Protocol (STP) and its variants are used to provide this. As of writing, I maintain a testbed in my home lab consisting of four (4) FS.com S3400 24-port 1000BASE-T PoE switches. These are used to exercise G.8032 support as per FS.com's documented test topology, in addition to its IEEE 802.1ag CFM requirements.

This project provides an interesting proof of concept for Layer 2 redundancy beyond STP (and, arguably providing an additional layer of network protection beyond proposed Layer 3 solutions such as Topology Independent Loop-Free Architecture (TI-LFA) from the IPv6 Segment Routing protocol suite, which has the shortcomings of having only sub-100ms target times for failover, and a hard dependence on Layer 3 routing state), however it does not go as far as IEEE 802.1Q-2022's Shortest Path Bridging (SPB) in its scope and intent.

V. Related work

A. Packet Construction Set

The Packet Construction Set (pcs), co-authored with George Neville-Neil, is currently being renovated for the post Python 3.11 world. Originally the unit test suite for FreeBSD IGMPv3/MLDv2 was developed using pcs, and it was used to great effect on my Ph.D. work on ILNP to gather Locator Update delay results for its dynamic re-routing capabilities across an IPv6 core network. Today, FreeBSD developers generally use Scapy or Packetdrill from Python scripting out of necessity, as pcs had bit-rotted over the years. I had added a significant number of features to pcs to support protocol-level conversations, including a Scapy-like syntax, match chain capabilities, and refactoring checksum handling. Particularly compelling is my addition of the cBPF module to handle BPF opcode streams, which could foreseeably extended for eBPF processing across platforms.

The new incarnation of pcs is being hosted at SourceHut, rather than GitHub. I would be happy to accept patches for Python "async" support; nanosecond resolution support for PCAP capture and dump files is in progress. Support for IPv6 Destination Options, and the associated ILNP Locator Update extension to ICMPv6 signalling, would need to be re-created, as they have been lost due to personal tragedy. The use of the PCAP-NG TLV format to implement indexing for large packet captures, e.g. using log-structured merge trees (LSM) would be a compelling future addition which might forseeably be adopted by Wireshark and other libpcap consumers. Michael Tuexen has suitable standardization efforts for the PCAP-NG format in progress as IETF WG I-Ds to support such use.

B. KScope Source Editor

Many FreeBSD developers have made use of the KScope C/C++ cross-referencing code editor over the years, however, it has since bit-rotted. I retain contact with its original author, Elad Lahav, who is Principal Engineer at QNX in Ottawa, Canada. I was introduced to it in the 00s by Prof. Robert Watson, when it was based on the KDE 3.x framework. I used Gilles Allard's KDE 4.x port to great effect on my Ph.D. work; he has since entirely deleted his GitHub account and no longer works on KScope, having migrated it back to SourceForge.

Since January 2026, I have undertaken to port KScope as it currently exists to KDE Frameworks 6. As of writing, 90% of the code is building with Qt5/KF5 with some changes to incorporate the use of C++11 and other newer language features; the code mostly remains based on C++03. Having evaluated Sourcegraph's open search engine and its uses, there is still clearly a case to be made for keeping KScope around, due to Sourcegraph's commercial nature and other limitations (e..g the difficulty of inspecting deep C/C++ call graphs within a browser-based application's history). However, KScope remains dependent on KDE because it is tied to KTextEditor in its implementation.

It is possible that an LSP-based language server, and appropriate IDE, could now replace KScope, in some cases. For the FreeBSD "kernel hacking" use case, there is probably no viable alternative; e.g. for CLion/IDEA, the JetBrains IDE

indexing & cross-referencing capabilities are very much tied to their “compilation database”, which is usually generated by clang/LLVM ecosystem tools. FreeBSD packages for KScope will be offered on an ongoing basis. It will now only be offered for Linux as an AppImage on a best-effort basis.

C. XORP Reloaded

I was a core developer on XORP from 2004-2010, before and during the XORP, Inc. privatized spin-out from the International Computer Science Institute (ICSI). XORP was re-released under a combination of GPL and LGPL licenses in the mid to late 00s. The company failed in early 2010, and the GPLed project on GitHub saw no real development traction over the last decade and a half.

The current owners of the XORP, Inc. IP portfolio appear to be Fiberstore (FS.com) based in Beijing. They acquired this third-hand through Pica8 after their bankruptcy, and in turn, Pica8 acquired it from Quanta (the OEM Sun Microsystems used for the Sun Fire series of servers, no doubt well known to Bryan Cantrill at Oxide Computer), and they are allegedly the buyers of the IP portfolio at the bankruptcy auction circa spring of 2010. Pica8 claim “ownership” of XORP in their marketing materials, mentioning the GPL, which is a little odd; they own the XORP, Inc. copyrights and presumably the IP package from the failed commercial venture, which is hardly the same thing as owning XORP itself. Whilst there is a source tarball available for a historic PicOS (formerly XorPlus) release, it has not been recently updated, and showed little or no signs of technical innovation. I did not inspect closely for the sake of preserving the notion of clean-room engineering.

The legacy XORP code base is key to my plans for Routeplex Limited. I plan to pursue a semi-commercial Delayed Open Source Publication (DOSP) based, organically grown startup strategy, leveraging the final BSD licensed release of XORP 1.5, and to pursue traditional copyright and trademark protection for the project, much as Jason Donenfeld has done for his WireGuard VPN technology. FreeBSD will be a key development target for testing TCP-AO enhancements, modulo future SRv6 support.

I also intend to make good on the promise of the original XORP architecture, which can be considered a broken one: a platform isn’t "eXtensible" if one has to write a significant amount C++ to interface with it. A concrete strategy is in place to address this significant source of “techncial debt”. For individuals, small scale edge routing, home labs, and IXP route reflectors, use cases for which ExaBGP or BIRD are often employed today, I would prefer to make a full open source release without encumberance. However, key IP components (e.g. the future BGP implementation) may need to be protected with the Mozilla Public License 2.0, if copyrighting APIs is internationally insufficient beyond protections at English law.

D. Google Summer of Code Student Supervision

1) Kernel/EFI Boot Counter for Early-Boot HMAC: The FreeBSD base system does not have a persistent reboot counter. This requirement became apparent during my Ph.D. work on ILNP, where the synchronization protocol used between distributed site routers required HMAC-SHA2-256 authentication from early boot. Many network protocols rely on message authentication based on Hash-based Message Authentication Code (HMAC). During early boot, inputs to these HMACs may be deterministic if sufficient entropy is not yet available. The UEFI firmware’s MonotonicCounter service provides an independent source of truth for the boot count, which may be used to seed HMAC. As of writing, we have two (2) viable student candidates for this project.

2) if_bridge(4) IGMP/MLD multicast snooping support: This feature is desirable to constrain the "flooding" of Ethernet frames to member ports as per IEEE 802.1D MAC Bridges specification. I am the original author of IGMPv3 and MLDv2 so I am best placed to supervise; arguably this is my central FreeBSD code contribution, and instrumental in achieving my Ph.D scholarship placement under Cisco’s University Research Program (URP) at the University of St Andrews in 2011. Sadly, as of writing, we do not yet have any viable candidates for this project.

E. Ongoing intellectual property issues in the era of Large Language Models (LLMs)

All my projects external to FreeBSD itself have been migrated to SourceHut as an intellectual property (IP) risk management measure. Related to the ongoing IP issues is the new position advocated by Bruce Perens, formerly of the Open Source Initiative (OSI), who is now promoting “PostOpen” as an alternative.

Drew DeVault has publicly committed to protecting the interests of open source and “PostOpen” developers alike. GitHub are complicit in unwarranted IP theft with OpenAI and litigation is ongoing. I have also observed that the Internet Archive’s use of LCP DRM protection for their PDF files is a potent deterrent to unauthorized and unwanted 3rd party Generative AI training.

Whilst I do not fully agree with Perens’ position, this is an important development and must be followed. I will continue to make my FreeBSD work available under the traditional BSD 2 and 3-clause licenses where rational self-interest permits this, however I now reserve the right to adopt the additional copyright protections offered at English law where this makes sense, in addition to pursuing non-OSI-approved, source-available alternative licensing schemes for my work outside of FreeBSD itself.

The OSI will not approve licenses that restrict AI training because they (arguably) violate the “no discrimination” clauses 5 and 6 of the Open Source Definition (OSD). It remains to be seen whether or not this definition holds up when subject to legal scrutiny. It is also quite likely that the OSD will be further contested in future. For now, Perens advises not to use the “open source” nomenclature.

F. Unusual gTLD domain name activity

My newly incorporated business name Routeplex, under which I intend to fold much of my ongoing open source work, is currently in use by a foreign entity who registered

routeplex.com in suspicious, bad faith circumstances in January. So-called “AI Slop” content appeared on the domain shortly afterwards. I retain control of the routeplex.co.uk and routeplex.net domains as hosted by Cloudflare. I have retained the legal option of obtaining a UK trademark and pursuing action against the foreign entity although I do not plan to do this in the immediate future, due to the time & business expense involved. The brand dilution problem, however, continues to exist. Please contact me out-of-band if interested in the circumstances of this hostile action.

VI. Future work

It is my strong belief that RFC 6740+ ILNP can only realistically be deployed with further research in addition to the previous work by myself and Gregor Haywood, and the ongoing work by Saleem Bhatti & Rodney Grimes. Moreover, that a DOSP strategy be pursued semi-commercially to protect this effort from unwarranted corporate interference, as there is a significant threat to existing, entrenched business models, should ILNP be deployed Internet-wide. Dr. Radia Perlman sadly encountered such issues with the adoption of the TRILL protocol, and this is a situation I am personally keen to avoid.

Routeplex Limited seeks to be at the forefront of this innovation, without venture capital involvement (modulo traditional “Angel” investor equity funding & debt funding). Although much of my doctoral work was previously funded by Cisco Systems, Inc, it was not published in journals or conferences due to time constants, and I retain copyright and IP ownership of the original code artefacts. Pursuant to this, I am tracking IETF SPRING WG activity on IPv6 Segment Routing closely, and the ANRW IRTF ’17 work on the Linux host-based implementation of SRv6. Further details about this are Commercial-in-Confidence and subject to a non-disclosure agreement (NDA).

Acknowledgment

With thanks to Daniel Loizos as an interim reviewer for this work proposal.

Bruce Simpson Domain expert in IP networking and telecommunications. Significant pro bono open source software development contributions dating back to 2003. Member of ACM and USENIX. Full professional CV available on LinkedIn.