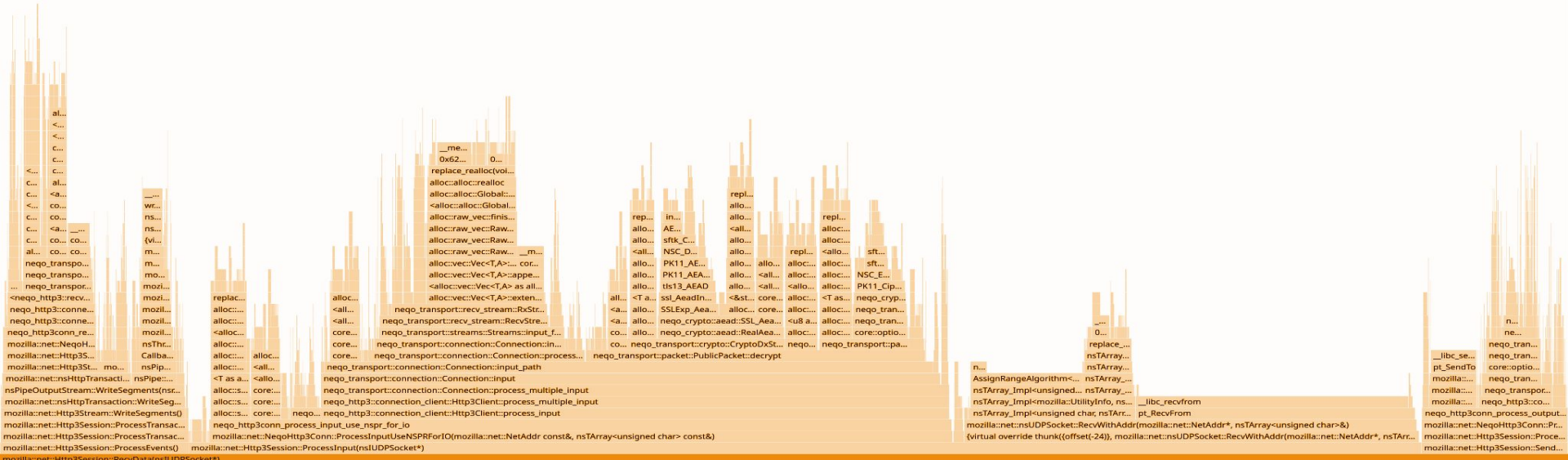
Fast UDP makes QUIC quicker

optimizing Firefox’s HTTP3 IO stack

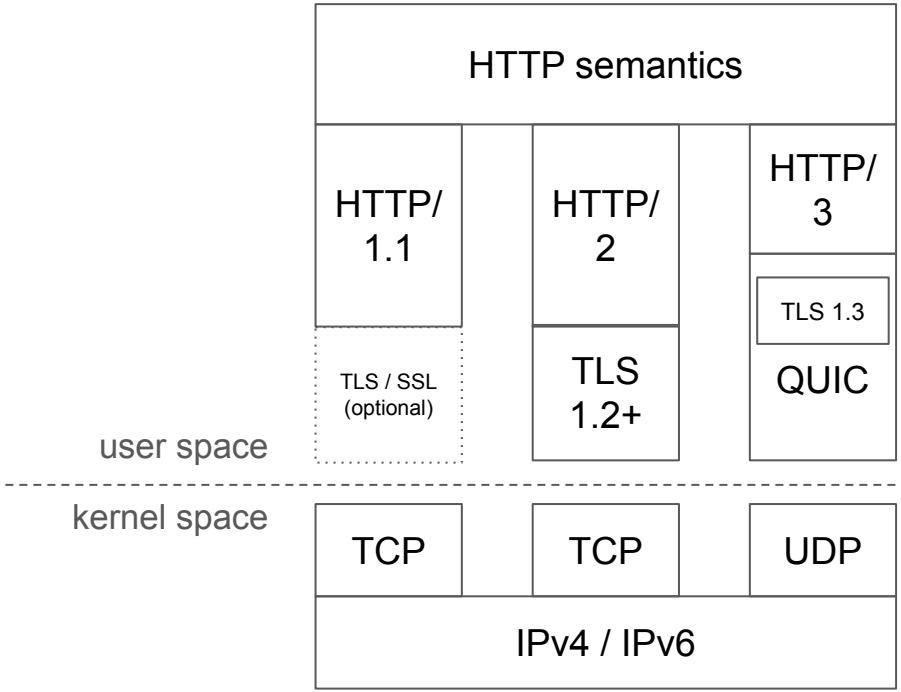
MIR^3 2025 - Max Inden



Max Inden

- Software engineer at Mozilla
- Working on HTTP3 and QUIC in Firefox
- mail@max-inden.de
- @mxinden
- Previously p2p, Kubernetes and Prometheus

QUIC runs on top of UDP



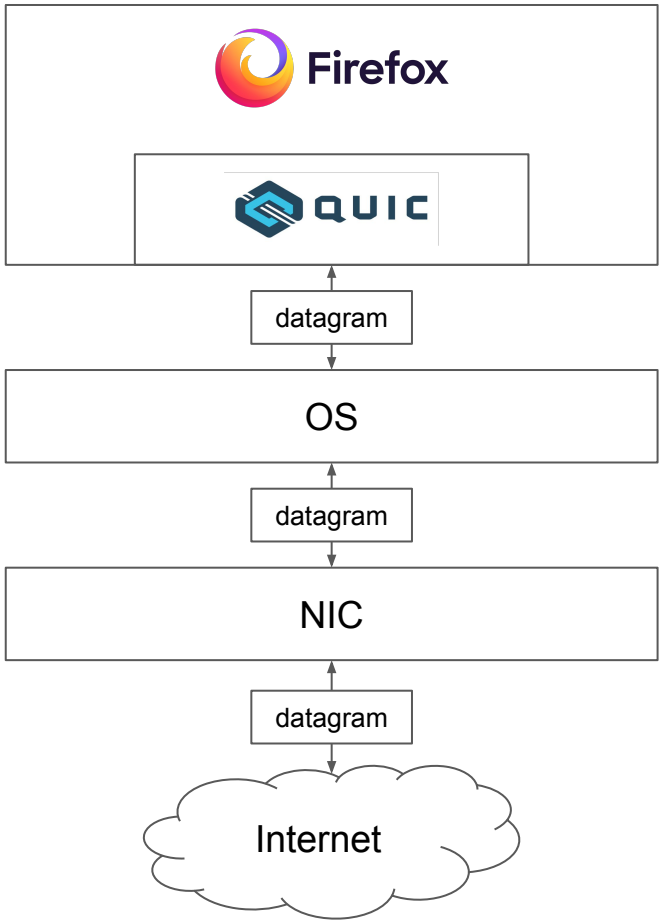
https://en.wikipedia.org/wiki/File:HTTP-1.1_vs._HTTP-2_vs._HTTP-3_Protocol_Stack.svg

QUIC in userspace

Unsurprisingly an un-optimized userspace transport protocol on top of UDP is not as fast as heavily optimized TCP in kernel space + NICS.

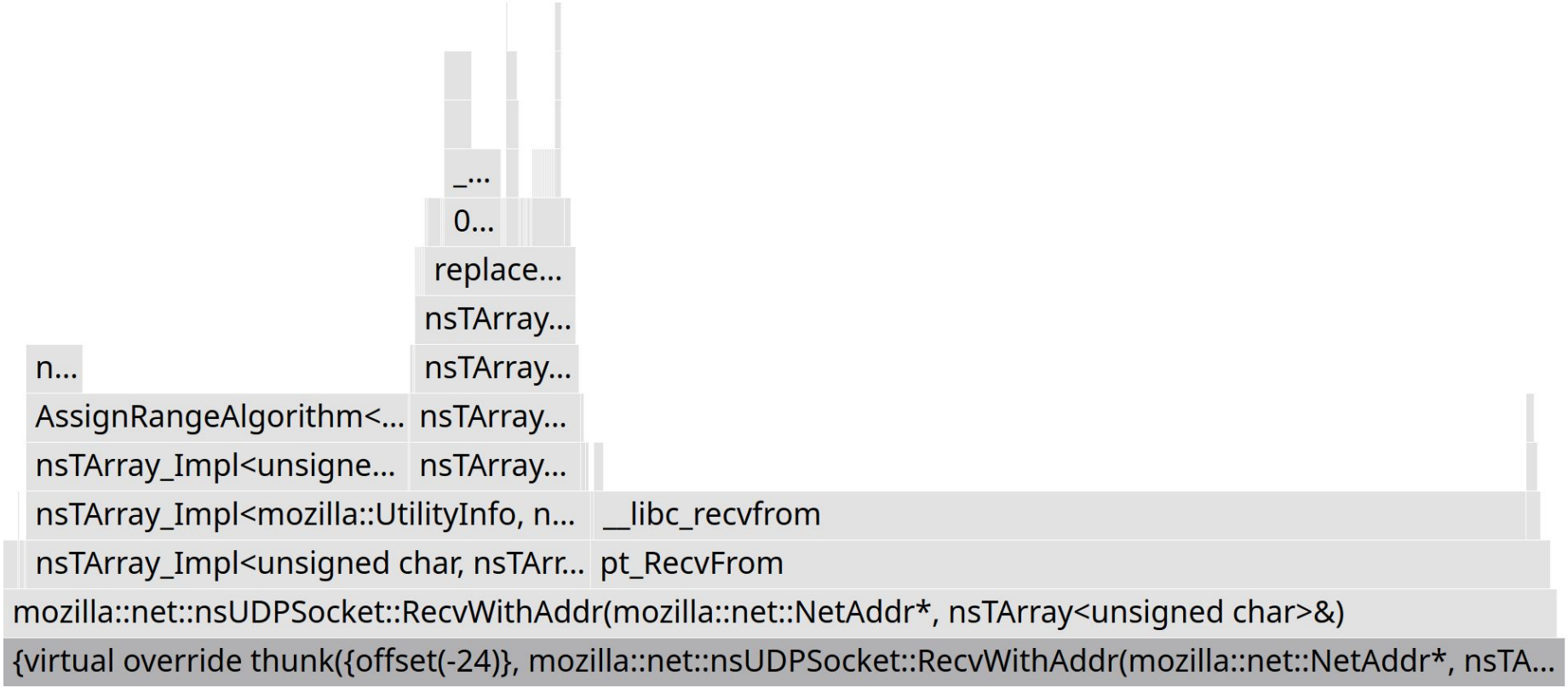
Userspace QUIC might do:

- one syscall per UDP datagram
- <= 1500 byte MTU for Internet traffic
- ACK of up to every second packet
- de Bruijn et al. “3.5x the CPU cycles per byte”



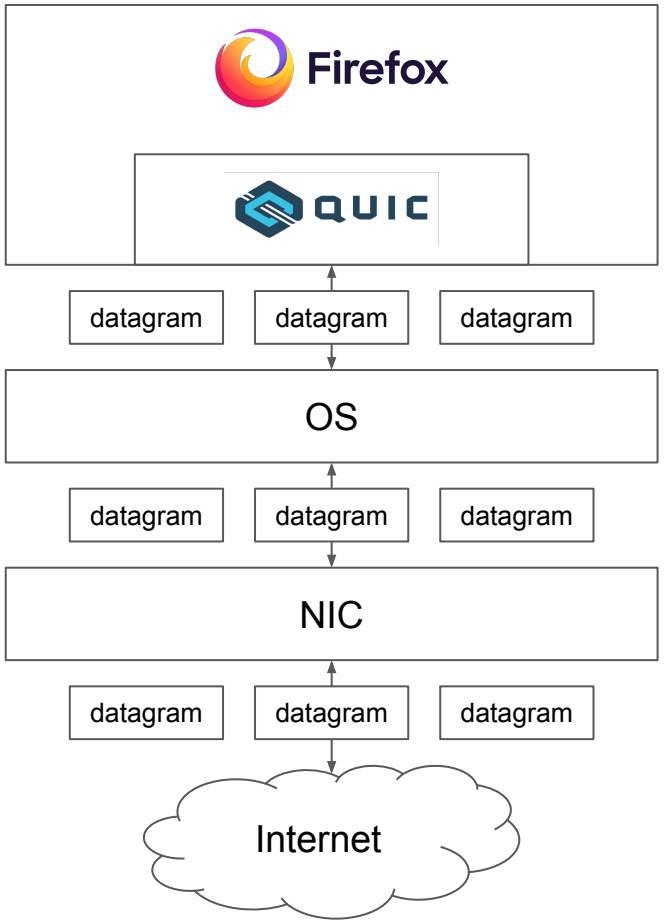
Un-optimized Firefox QUIC UDP IO

Un-optimized Firefox QUIC UDP IO



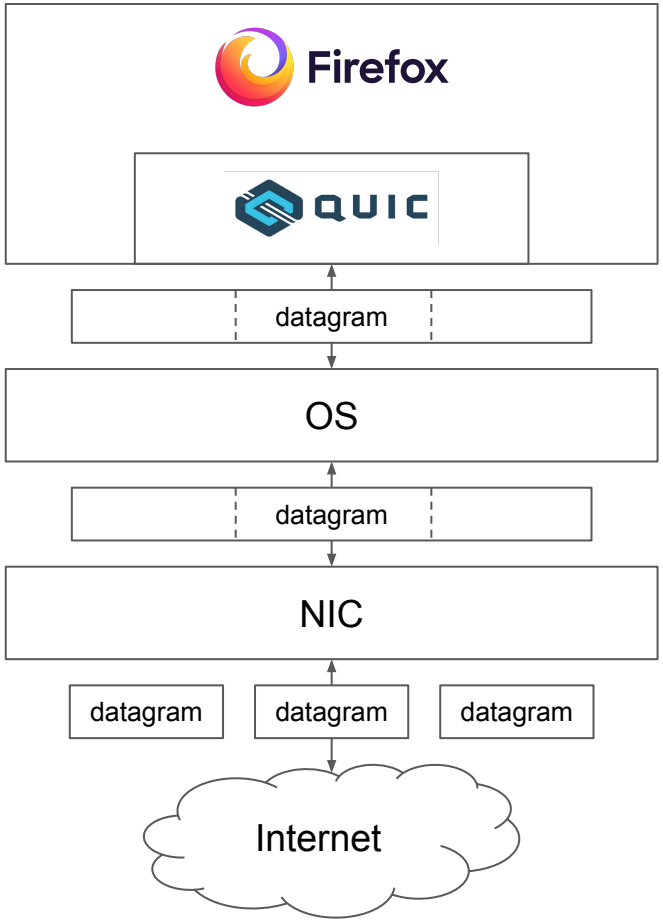
Multi-message syscalls

- MacOS sendmsg_x and recvmsg_x
- Linux's sendmmsg and recvmmsg
- 11% performance improvement on loopback transfer



Segmentation offloading

- Linux's GSO/GRO
- Windows' USO/URO
- measurements on Firefox Nightly Linux GRO
  - 75th of read syscalls read 2 or more packets, 95th read 10 or more packets.
  - 75th of read syscalls read 2.4 KiB total, 95th read 12 KiB total.
- close to the 4 Gbit/s on loopback benchmark



Windows

- WSASendMsg and WSARecvMsg
- > fosstodon.org doesn't load with network.http.http3.use_nspr_for_io=false on ARM64 Windows
- URO disabled for now
  - On some drivers no segment size
- USO disabled for now
  - High packet loss
  - Crashes some drivers

MacOS

- sendmsg and recvmsg
- sendmsg_x recvmsg_x
- Apple network framework

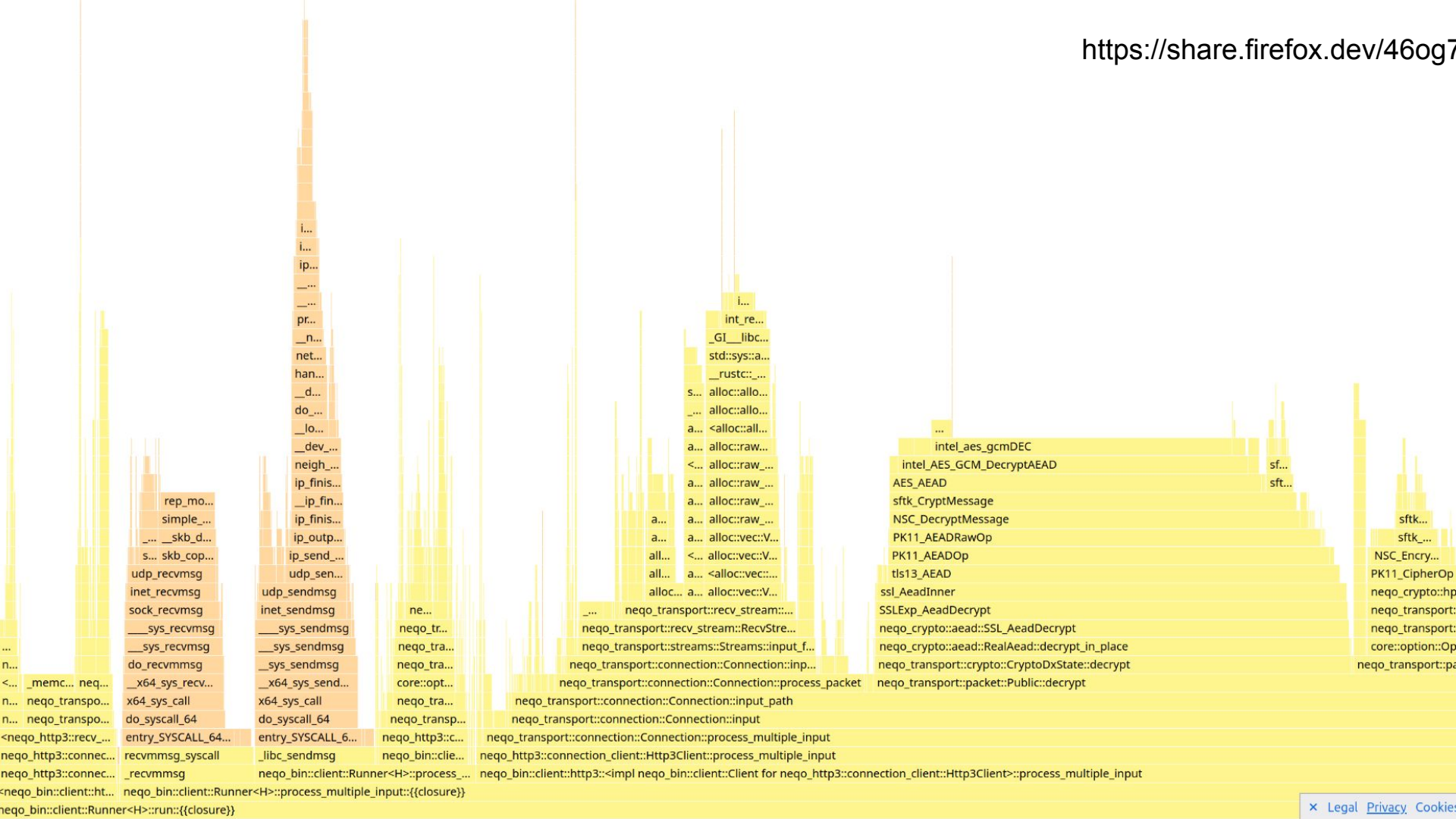
Linux

- sendmsg and recvmsg
- No mmsg system calls
  - One socket per connection
  - Segmentation offloading superior to mmsg
- GSO (max 10 datagrams) and GRO (64k buffer)
-

Android

- sendmsg and recvmsg
- No mmsg system calls
  - One socket per connection
  - Segmentation offloading superior to mmsg
- GSO (max 10 datagrams) and GRO (64k buffer)

- But Android is not Linux


https://share.firefox.dev/46og7
i...
i...
ip...
__...
__...
pr...
__n...
net...
han...
__d...
do_...
__lo...
__dev_...
neigh_...
ip_finis...
__ip_fin...
ip_finis...
ip_outp...
ip_send_...
udp_sen...
rep_mo...
simple_...
__skb_d...
skb_cop...
udp_recvmsg
inet_recvmsg
sock_recvmsg
___sys_recvmsg
___sys_recvmsg
do_recvmmsg
__x64_sys_recv...
x64_sys_call
do_syscall_64
entry_SYSCALL_64...
recvmmsg_syscall
_recvmmsg
udp_sendmsg
inet_sendmsg
___sys_sendmsg
___sys_sendmsg
__sys_sendmsg
__x64_sys_send...
x64_sys_call
do_syscall_64
entry_SYSCALL_6...
_libc_sendmsg
neqo_bin::client::Runner<H>::process_...
neqo_bin::client::Runner<H>::process_multiple_input::{{closure}}
neqo_bin::client::Runner<H>::run::{{closure}}
ne...
neqo_tr...
neqo_tra...
neqo_tra...
core::opt...
neqo_tra...
neqo_transp...
neqo_http3::c...
neqo_bin::clie...
i...
int_re...
_GI__libc...
std::sys::a...
__rustc::_...
s... alloc::allo...
_... alloc::allo...
a... <alloc::all...
a... alloc::raw...
<... alloc::raw_...
a... alloc::raw_...
a... alloc::raw_...
a... a... alloc::raw_...
a... a... alloc::vec::V...
all... <... alloc::vec::V...
all... a... <alloc::vec::...
alloc... a... alloc::vec::V...
_... neqo_transport::recv_stream::...
neqo_transport::recv_stream::RecvStre...
neqo_transport::streams::Streams::input_f...
neqo_transport::connection::Connection::inp...
neqo_transport::connection::Connection::process_packet
neqo_transport::connection::Connection::input_path
neqo_transport::connection::Connection::input
neqo_transport::connection::Connection::process_multiple_input
neqo_http3::connection_client::Http3Client::process_multiple_input
neqo_bin::client::http3::<impl neqo_bin::client::Client for neqo_http3::connection_client::Http3Client>::process_multiple_input
...
intel_aes_gcmDEC
intel_AES_GCM_DecryptAEAD
AES_AEAD
sftk_CryptMessage
NSC_DecryptMessage
PK11_AEADRawOp
PK11_AEADOp
tls13_AEAD
ssl_AeadInner
SSLExp_AeadDecrypt
neqo_crypto::aead::SSL_AeadDecrypt
neqo_crypto::aead::RealAead::decrypt_in_place
neqo_transport::crypto::CryptoDxState::decrypt
neqo_transport::packet::Public::decrypt
sf...
sft...
sftk...
sftk_...
NSC_Encry...
PK11_CipherOp
neqo_crypto::hp
neqo_transport:
neqo_transport:
core::option::Op
neqo_transport::pa
Legal
Privacy
Cookies

ECN

- Mark and report ECN
- Groundwork for L4S
- ~50% of paths ECN capable
- 75th percentile of QUIC connections see >= 0.6% CE marks on receive path.

Additional wins

- QUIC UDP IO in Rust using quinn-udp
- optimized memory management
  - we use a single long-lived 64k receive buffer for all QUIC connections
  - soundness check at compile time via Rust's borrow checker
  - Inplace en-/decryption
  - significant reduction in CPU time on large transfers

Next steps

- USO and URO on Windows
- Rollout on Android

max-inden.de/post/fast-udp-io-in-firefox/