Blitzping AF_XDP Enhancement Project

This project extends the open-source Blitzping tool with AF_XDP socket support and hop-by-hop analysis capabilities. It aims to provide high-performance packet processing with minimal latency.

Project Overview

Update Blitzping

Implement AF_XDP socket support for high-performance packet processing.

Hop-by-Hop Analysis

Build node and delay information with minimal GUI.

Performance Optimization

Achieve higher throughput and lower latency than traditional sockets.

AF_XDP Socket Advantages

Traditional Socket

Application → Socket API → Protocol Stack → Generic Driver → NIC

Multiple data copies between layers:

- NIC to kernel memory
- Kernel to socket buffer
- Socket to userspace

AF_XDP Socket

Application → AF_XDP Socket → XDP Program → Driver → NIC

Direct path with key advantages:

- Zero-copy operation
- Bypass protocol stack
- Memory mapped buffers
- Reduced context switching

Technical Implementation

UMEM Setup

Create memory region for zero-copy packet handling with aligned allocation.

Ring Buffer Configuration

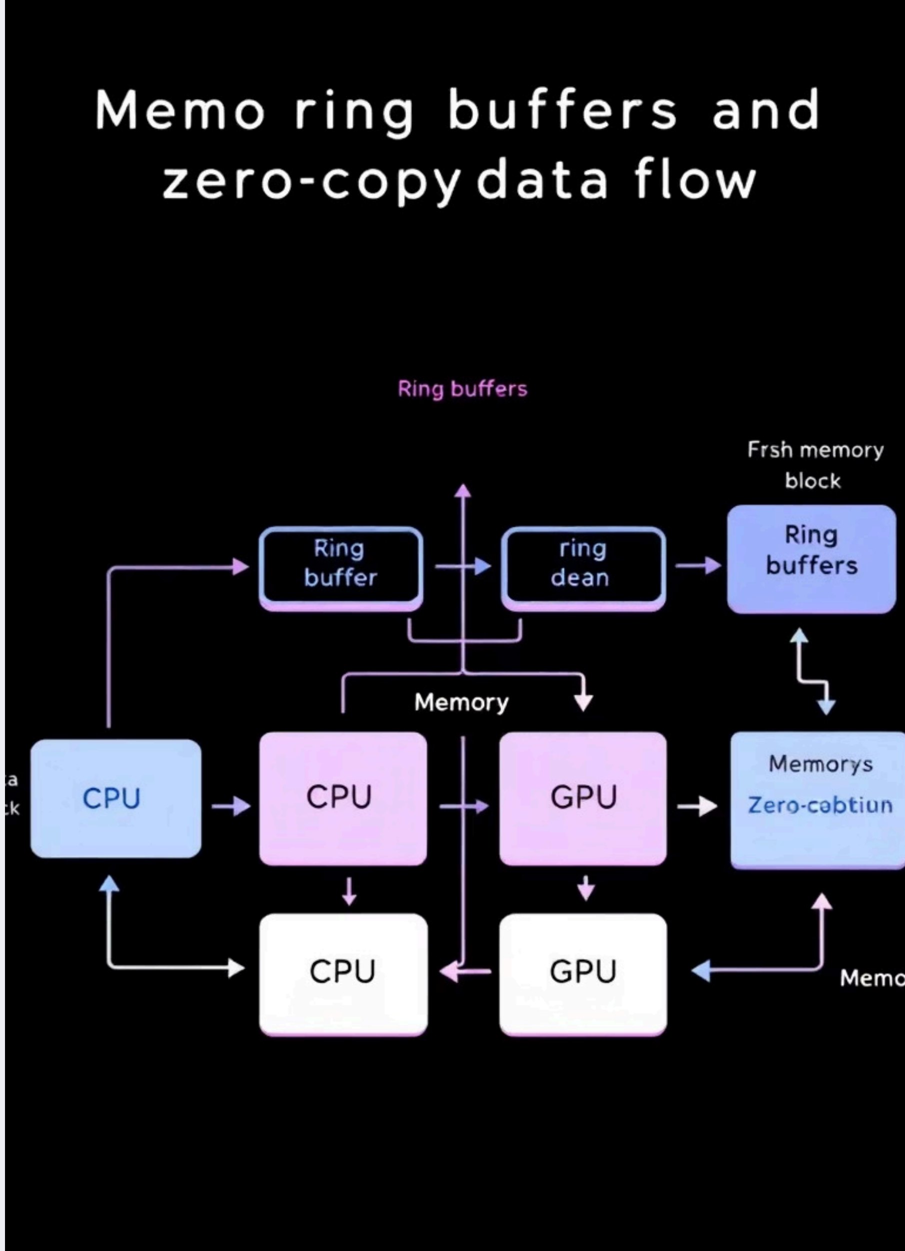
Configure optimal ring sizes for TX/RX and completion rings.

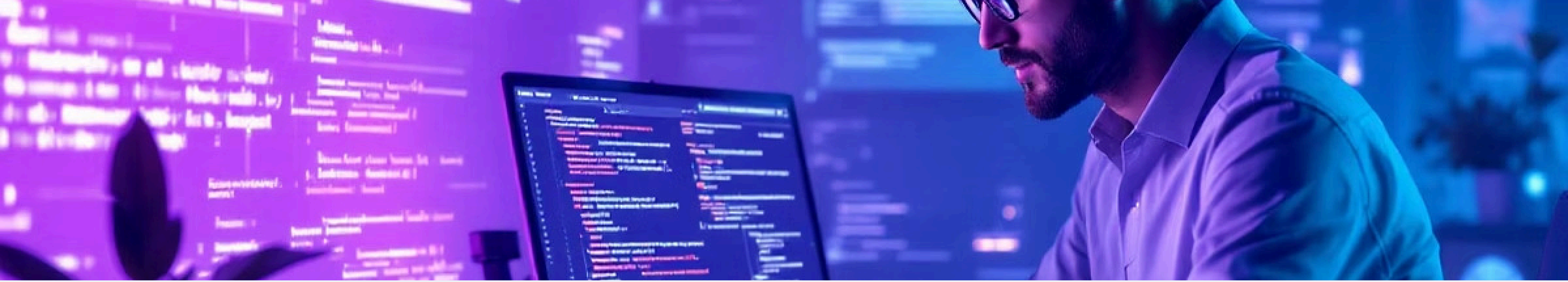
Memory Mapping

Map rings to userspace memory for direct access.

Interface Binding

Bind socket to specific interface and queue ID.





XDP Program Integration

Write XDP Program

Create BPF-C program for packet processing.

Compile Program

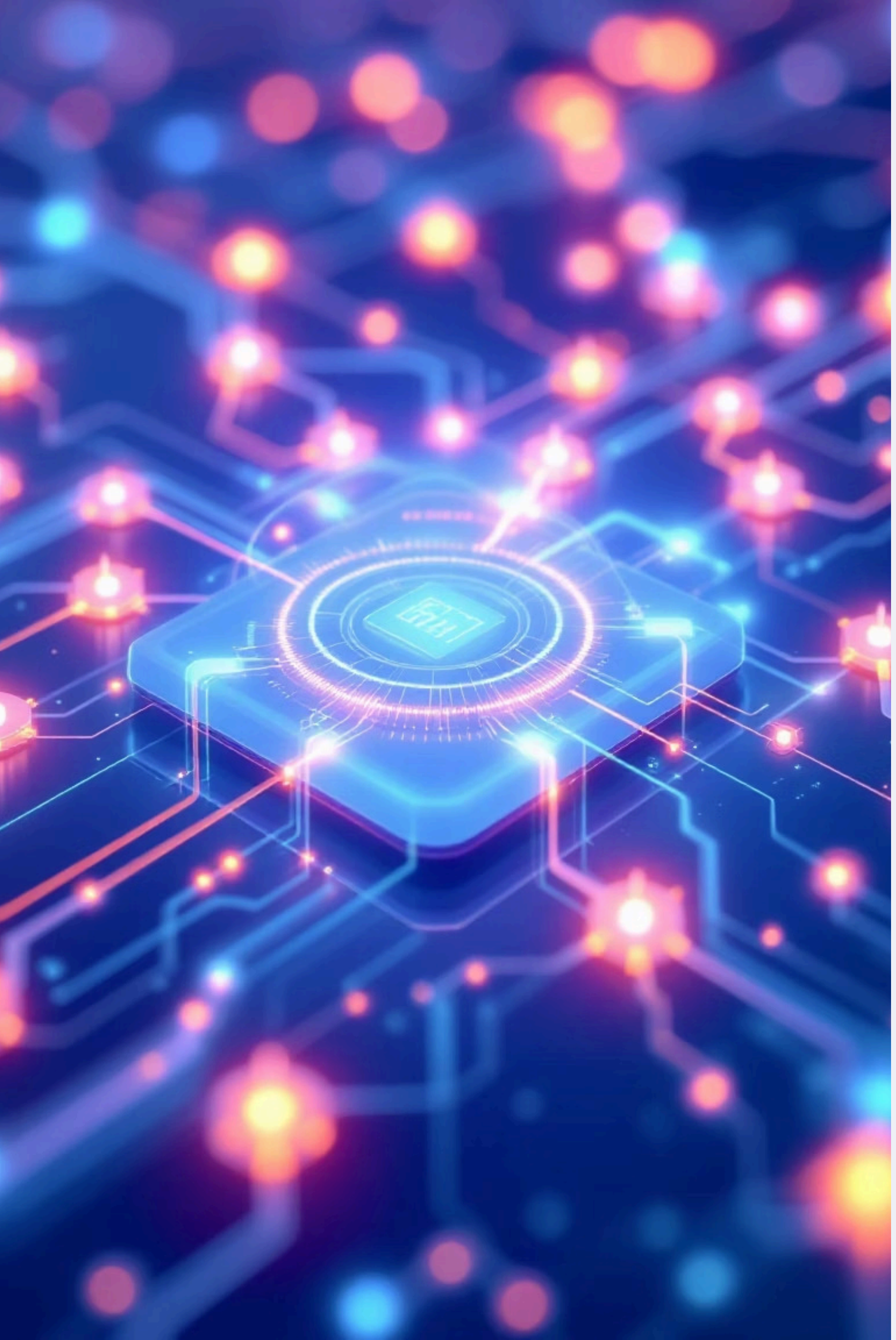
Use clang to compile for BPF target.

Load Program

Attach to interface with ip link command.

Update BPF Map

Connect socket reference to program.



Advanced Integration Examples

XDP-based Load Balancer

Distribute packets across multiple queues with custom XDP program. Run multiple Blitzping instances on different queues.

Packet Filter + Blitzping

Filter specific traffic with XDP program. Process only filtered packets with Blitzping in process mode.

Hardware Offload Integration

Enable hardware offload if supported by NIC. Run Blitzping with appropriate settings.

Performance Optimization Tips

CPU Pinning

Use taskset to pin processes to specific CPU cores.

Huge Pages

Reduce TLB misses with large-page memory allocation.

Ring Buffer Tuning

Adjust ring sizes for optimal throughput.

NIC Configuration

Optimize driver settings with ethtool.

Troubleshooting Common Issues

Binding Failures

Check interface, queue ID, kernel version, and permissions.

Performance Issues

Verify driver XDP support, increase buffer sizes, use huge pages.

Redirection Problems

Confirm map updates, XDP attachment, and matching queue IDs.

Hop-by-Hop Delay Analysis

Set TTL Value

Control how far packet travels using setsockopt.

Send ICMP Request

Construct and send Echo Request with unique identifier.

Measure RTT

Calculate round-trip time between send and receive.

Identify Hop

Extract IP address from response and print hop information.

Working Example Results

Hop	IP Address	RTT (ms)
1	10.7.0.5	2.333
5	10.154.8.137	12.442
8	72.14.204.62	13.950
11	8.8.8.8	16.076