POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

NETWORK COMPUTING
Computer Science and Engineering
Ingegneria Informatica

Final Project 2024-25

Author:
Shreesh Kumar Jha

Student ID:
11022306

Advisor:
Prof. Gianni Antichi & Prof. Sebastiano Milano

Academic Year:
2024-25

Contents

1 — TCP Connection Tracker Analysis and Performance Report

1.1. Source Code Analysis

In this report and github repository, I analysed, tested, and debugged the simplified eBPF/XDP TCP connection tracker provided for this project. Originally a basic prototype that evolved into a high-performance system sustaining 3.57 Gbps TCP throughput, I focused on:

TCP State Machine & Handshake Logic: Identified and corrected cases where SYN/SYN-ACK/ACK transitions were mishandled—preventing half-open or simultaneous-open connections from slipping through.

RST/FIN Teardown Handling: Found and fixed missing or incorrect state updates on connection close/reset, ensuring torn-down connections no longer linger in the tracking map.

TTL-Based Garbage Collection: Verified that stale entries are purged reliably, uncovering edge cases where inactive flows persisted past their timeout and tightening expiry checks accordingly.

UDP Flow Tracking: Reviewed minimal UDP "flow" state logic, addressing scenarios where packet bursts could overflow tracking buckets or bypass timeout rules.

Based on this analysis, I implemented targeted fixes—correcting state transitions, closing teardown gaps, reinforcing TTL expiry, and hardening UDP timeouts—to ensure accurate state maintenance under multi-gigabit loads.

1.2. Critical Bugs Identi ed

1.2.1. Faulty TCP Flag Logic

Issue: Incorrect Boolean expressions prevented SYN-ACK/ACK detection. **Impact:** Broken three-way handshake.

1.2.2. No RST/FIN Handling

Issue: No connection cleanup on termination. **Impact:** “Zombie” entries persisted indefinitely.

1.2.3. No Garbage Collection

Issue: Connections never expired. **Impact:** Map exhaustion under churn.

1.2.4. UDP Not Supported

Issue: All UDP packets dropped. **Impact:** Single-protocol limitation.

1.3. Corner Cases

Handshake Progression: SYN_SENT   SYN_RECV   ESTABLISHED transitions failed

Connection Persistence: No cleanup mechanism for completed connections

Protocol Limitations: TCP-only support limiting deployment

1.4. Implemented Fixes

1.4.1. TCP Flag Detection

```
// SYN detection
if (pkt.flags == TCPHDR_SYN) { /* Insert SYN_SENT */ }

// SYN+ACK detection
if ((pkt.flags & (TCPHDR_SYN | TCPHDR_ACK)) == (TCPHDR_SYN |
   TCPHDR_ACK)) {
    /* Transition SYN_SENT -> SYN_RECV */ }

// ACK detection
if (pkt.flags == TCPHDR_ACK) { /* Transition SYN_RECV ->
   ESTABLISHED */ }
```

Listing 1.1: Corrected flag logic

Result: Complete three-way handshake functionality.

1.4.2. RST & FIN Handling

```
// RST handling
if (pkt.flags & TCPHDR_RST) {
    bpf_map_delete_elem(&connections, &key);
    goto PASS;
}

// FIN state transitions
if (saved_state == ESTABLISHED && (pkt.flags & TCPHDR_FIN)) {
    v->state = FIN_WAIT_1;
}
if (saved_state == LAST_ACK && pkt.flags == TCPHDR_ACK) {
    bpf_map_delete_elem(&connections, &key);
}
```

Listing 1.2: Connection cleanup

Result: Proper connection termination per RFC 793.

1.4.3. TTL-Based Garbage Collection

```
// Expire check on lookup
if (v && v->ttl < now) {
    bpf_map_delete_elem(&connections, &key);
    v = NULL;
}

// Set TTL on creation
newEntry.ttl = now + TCP_SYN_SENT_TIMEOUT; // 2 minutes
```

Listing 1.3: Automatic cleanup

Timeouts: TCP SYN (2min), ESTABLISHED (5 days), FIN (2min), UDP (5-10min).

1.4.4. UDP Flow Tracking

```
// New UDP flow
newEntry.state = NEW;
newEntry.ttl = now + UDP_FLOW_TIMEOUT;

// Bidirectional detection
if ( same_dir) {
    v->state = ESTABLISHED; // Promote to bidirectional
}
```

Listing 1.4: UDP support

Result: 817 Mbps UDP throughput at 1 Gbps o ered (4% loss).

1.5. Performance Metrics

Performance Metric	Original	Optimized	Improvement
TCP Throughput (Gbps)	0.283	3.57	12.6x
Connection Stability	1 sec	10+ sec	10x
Packet Retransmissions	5 per test	0	100% reduction
UDP Throughput (Mbps)	0	817	New capability
Protocol Support	TCP only	TCP + UDP	2x protocols

Table 1.1: Comprehensive performance comparison between original and optimized implementations

1.6. Conclusion

The analysis identified four critical bugs preventing optimal functionality and successfully implemented comprehensive fixes. Key achievements:

Performance: Achieved 3.57 Gbps TCP throughput (12.6x improvement)

Protocol Support: Added UDP with 817 Mbps capability

Resource Management: TTL-based garbage collection prevents map exhaustion

Compliance: RFC 793-compliant TCP state machine implementation