Software Requirements Specification

Project Title: AegisFlow

Sub-title: Semi-Autonomous Kernel-Level Traffic Orchestrator for Crisis and Enterprise Networks

Version: 1.0 (Architecture & Prototype Specification)

Lead: Ram Gour

1. Executive Summary

Modern networks fail not because they stop functioning, but because they become indiscriminate under stress. When bandwidth collapses—due to disaster, attack, or overload—critical signals compete equally with non-essential traffic. Encryption removes payload visibility, and human operators cannot react at packet timescales.

AegisFlow is a kernel-resident traffic orchestration system designed to preserve service continuity for mission-critical communications during extreme congestion or misuse. It operates at the network edge and classifies traffic based on behavioral characteristics (timing, size, burst patterns, flow dynamics) rather than payload inspection. Based on this classification and operator-defined policy, AegisFlow enforces real-time prioritization, throttling, or suppression of traffic classes.

The system combines:

- In-kernel, wire-speed telemetry and enforcement using eBPF/XDP
- Behavioral machine learning for encrypted traffic classification
- Policy-driven control with human override
- Real-time visualization and auditability for operational trust

AegisFlow is designed for deployment at edge gateways in government, ISP, enterprise, and academic environments where availability under stress is more important than best-effort fairness.

2. Problem Statement

2.1 Crisis Networks

In disasters, physical infrastructure is often partially destroyed. Remaining links become saturated by a mix of life-safety telemetry, human communications, and high-bandwidth non-essential uploads. Because most traffic is encrypted, traditional firewalls and DPI systems cannot distinguish importance, only origin or port. Manual intervention is slow, error-prone, and does not scale.

The result is that critical signals are delayed or dropped due to congestion caused by non-critical traffic.

2.2 Enterprise and Institutional Networks

In universities and enterprises, limited up-links are frequently saturated by streaming media, large downloads, and background synchronization. This degrades performance of business-critical APIs, research platforms, examination systems, and collaboration tools. Static QoS rules based on ports or IP addresses are increasingly ineffective due to encryption and protocol multiplexing.

3. Goals and Design Principles

3.1 Primary Goals

- Maintain high availability for critical traffic under congestion.
- Operate independently of payload visibility (encryption-agnostic).
- Enforce decisions at kernel or NIC level with minimal latency.

- Support semi-autonomous operation with human oversight.
- Provide audit-ability and explain-ability for decisions.

3.2 Design Principles

- Behavior over identity: classify flows by how they behave, not by origin or content.
- Fast path stays simple: packet path logic must remain deterministic, bounded, and verifiable.
- Control is policy-driven: AI suggests, policy decides, kernel enforces.
- Human remains sovereign: operators can override, pin, or exempt flows at any time.
- Fail safely: in uncertainty, degrade gracefully rather than catastrophically.

4. System Overview

AegisFlow is composed of four cooperating subsystems:

1. Data Plane (Kernel Space): High-performance eBPF/XDP programs attached at the NIC ingress path collect flow telemetry and apply enforcement actions such as pass, mark, rate-limit, or drop.
2. Control Plane (User Space): A management service orchestrates feature extraction from flow telemetry, machine learning inference, policy evaluation, and updates to kernel maps and enforcement rules.
3. Intelligence Layer: A behavioral classification engine trained on flow-level features such as packet size distributions, inter-arrival time statistics, throughput and burstiness, and flow duration and directionality.
4. Operations Interface (HITL Dashboard): A real-time visual and control interface providing live network and flow state visualization, classification outcomes and confidence, policy actions and overrides, and decision provenance.

5. Target Deployment Environments

5.1 Edge Gateways (Primary Target)

- ISP regional routers
- Mobile network aggregation points
- Disaster-response up-links
- Enterprise or campus border gateways

5.2 Use Case Categories

A. Crisis Management (Government / ISP): Severe bandwidth reduction due to infrastructure damage. Objective: preserve life-safety and coordination traffic by suppressing non-essential bulk traffic automatically and immediately.

B. Enterprise / Academic Networks: Chronic or peak-time congestion due to entertainment and bulk transfers. Objective: ensure predictable performance for business, research, and operational systems by dynamically constraining low-priority flows.

6. Functional Requirements

FR-01: Kernel-Level Interception. The system shall intercept packets at the NIC ingress path using XDP to avoid traversal of the standard Linux network stack.

FR-02: Flow Identification. The system shall identify and track traffic using flow keys derived from L3/L4 headers.

FR-03: Behavioral Telemetry Collection. The system shall maintain per-flow metrics including at minimum: packet count, byte count, first-seen and last-seen timestamps, inter-arrival time statistics, directional counters, and flow duration indicators.

FR-04: Feature Derivation. The control plane shall compute derived features such as average and variance of packet size, throughput, burstiness, temporal regularity, and flow asymmetry ratios.

FR-05: Behavioral Classification. The system shall classify flows into policy-relevant categories using machine learning models trained on metadata-only features.

FR-06: Policy Evaluation. The system shall evaluate classification results against operator-defined policies to determine enforcement actions.

FR-07: Enforcement Actions. The system shall support enforcement actions including priority marking, rate limiting, throttling, and dropping or blocking flows.

FR-08: Human Override (HITL). The system shall allow operators to manually override classification or policy decisions for any active flow.

FR-09: Real-Time Visualization. The system shall provide a real-time visualization of network state, flows, and applied policies.

FR-10: Decision Provenance and Audit. The system shall record and present the features influencing a classification, the policy rule applied, and the resulting enforcement action.

7. Non-Functional Requirements

NFR-01: Performance. The data plane shall operate with near-native throughput and sub-millisecond added latency.

NFR-02: Determinism in the Fast Path. Kernel-level packet processing shall avoid unbounded loops, dynamic memory allocation, or variable-time operations.

NFR-03: Scalability. The architecture shall support scaling to large numbers of concurrent flows and multi-node deployments in future iterations.

NFR-04: Resource Efficiency. The system shall operate within constrained environments, including edge devices and virtualized test-beds.

NFR-05: Reliability and Fail-Safe Behavior. In the event of control-plane or ML subsystem failure, the data plane shall continue operating in a predefined safe policy mode.

NFR-06: Observability. All major system decisions and state transitions shall be observable and logged.

8. Technical Architecture

- Data Plane: eBPF / XDP (C)
- Control Plane: Python with BCC, FastAPI, Socket-based event streaming
- Intelligence Layer: Random Forest or XGBoost classifiers
- Simulation Environment: Mininet and Open vSwitch
- Visualization and Control: Three.js, Gradio, Daggr, web-based dashboard
- Operating System: Linux Kernel 6.x or later

9. Validation Strategy

- Synthetic traffic generation for multiple behavioral classes
- Congestion injection in simulated topologies
- Measurement of classification accuracy, latency impact, throughput under stress, and policy enforcement effectiveness
- Operator-in-the-loop scenario testing

10. Strategic Positioning

AegisFlow aligns with proven industry patterns: kernel-level packet processing, edge-based enforcement for latency and scalability, behavior-based traffic analysis for encrypted networks, and automation with human supervisory control. The system is designed not as a firewall replacement, but as a resilience layer for networks under stress.

11. Scope of Version 1.0

Version 1.0 focuses on single-node deployment, flow-level behavioral classification, policy-driven prioritization and throttling, simulation-based validation, and a human-in-the-loop operational interface. Distributed control and carrier-scale integration are explicitly out of scope for this version.