

# NetSim Refactoring and Extension Plan

Enzo J

## 1 Packet Transmission Simulation

**Current State:** Implements basic Poisson process, but lacks clear demonstration in output.

**Refactoring Needed:**

- a. Enhance Poisson process implementation to more accurately model packet arrivals.
- b. Add detailed logging of packet generation times to demonstrate Poisson distribution.
- c. Implement variable packet sizes based on realistic network scenarios.

## 2 Bandwidth Utilization

**Current State:** Calculates basic utilization, but lacks dynamic behavior.

**Refactoring Needed:**

- a. Implement time-varying bandwidth capacity to simulate network fluctuations.
- b. Add congestion-aware bandwidth calculation.
- c. Incorporate QoS considerations in bandwidth allocation.

## 3 Network Error Rate

**Current State:** Uses signal-to-noise ratio, but lacks sophistication.

**Refactoring Needed:**

- a. Implement more complex error models (e.g., burst errors, packet loss probability).
- b. Add environmental factors affecting error rates (e.g., interference, distance).
- c. Implement error correction and retransmission mechanisms.

## 4 Load Balancing

**Current State:** Uses simple distribution, lacks dynamic adjustment.

**Refactoring Needed:**

- a. Implement adaptive load balancing algorithms (e.g., least connections, round-robin).
- b. Add real-time monitoring and adjustment of load distribution.
- c. Incorporate network health metrics in load balancing decisions.

## 5 Network Congestion

**Current State:** Returns fixed congestion level, lacks realism.

**Refactoring Needed:**

- a. Implement dynamic congestion calculation based on current network state.
- b. Add congestion control mechanisms (e.g., TCP congestion control algorithms).
- c. Simulate effects of congestion on other network parameters (delay, throughput).

## 6 Exponential Network Delay

**Current State:** Implements basic delay, but doesn't clearly use exponential distribution.

**Refactoring Needed:**

- a. Refactor to clearly use exponential distribution for delay modeling.
- b. Implement jitter simulation.
- c. Add correlation between network load and delay.

## 7 Network Failures

**Current State:** Basic failure simulation, lacks complexity.

**Refactoring Needed:**

- a. Implement various failure modes (link failure, node failure, partial failures).
- b. Add cascading failure scenarios.
- c. Implement failure recovery mechanisms and measure recovery time.

## 8 Routing Efficiency

**Current State:** Returns fixed efficiency value, lacks dynamism.

**Refactoring Needed:**

- a. Implement multiple routing algorithms for comparison (e.g., Dijkstra's, Bellman-Ford).
- b. Add dynamic route recalculation based on network conditions.
- c. Implement metrics for routing overhead and convergence time.

## 9 Topology Change Impact

**Current State:** Returns fixed impact value, lacks realism.

**Refactoring Needed:**

- a. Implement dynamic topology generation and modification.
- b. Simulate impact of topology changes on all other network parameters.
- c. Add support for different network topologies (mesh, star, tree, etc.).

## 10 Packet Prioritization (WIP)

**Current State:** Basic implementation, not integrated into simulation.

**Refactoring Needed:**

- a. Complete implementation of packet prioritization algorithms.
- b. Integrate prioritization with other modules (e.g., congestion control, load balancing).
- c. Implement different QoS classes and their impact on network behavior.

## 11 General Improvements

**Refactoring Needed:**

- a. Implement a central simulation clock for synchronization of all modules.
- b. Add extensive logging and visualization capabilities for all modules.
- c. Implement a configuration system for easy parameter adjustment.
- d. Develop a comprehensive test suite using Google Test framework.
- e. Optimize performance for large-scale simulations using parallel processing.
- f. Implement inter-module dependencies and interactions for more realistic simulation.

## 12 Priority Order for Implementation

1. Central Simulation Clock (11a)
2. Dynamic Congestion Calculation (5a)
3. Adaptive Load Balancing (4a, 4b)
4. Enhanced Packet Transmission Simulation (1a, 1b)
5. Improved Network Delay Modeling (6a, 6b)
6. Dynamic Topology Changes (9a, 9b)
7. Realistic Error Rate Modeling (3a, 3b)
8. Dynamic Routing Efficiency (8a, 8b)
9. Complete Packet Prioritization (10a, 10b)
10. Advanced Network Failure Simulation (7a, 7b)
11. Comprehensive Logging and Visualization (11b)
12. Inter-module Dependencies (11f)
13. Configuration System (11c)
14. Test Suite Development (11d)
15. Performance Optimization (11e)

This refactoring plan addresses the discrepancies between the current implementation and the original vision, focusing on creating a more dynamic, realistic, and comprehensive network simulation.