Bayesian Agent Orchestrator (BAO) for Network Intrusion Detection

Introduction

- **The Context:** Modern Network Intrusion Detection Systems (NIDS) utilize heterogeneous detectors (ML classifiers, heuristic filters, anomaly detectors) but suffer from high false-positive rates and "alert fatigue".
- **The Flaw:** Current systems naively aggregate outputs using fixed thresholds or rigid cascades, ignoring asymmetric costs (e.g., missed attacks vs. wasted analyst time) and failing to quantify true epistemic uncertainty.
- **The Solution:** The Bayesian Agent Orchestrator (BAO) fuses detector alerts probabilistically, maintains a posterior belief over the hidden threat state, and makes cost-sensitive decisions (alert, defer, or acquire more evidence) using a Value of

Objectives

1. Orchestration Layer: Implement a Bayesian control layer that treats LLMs and tools as black-box observation sources. It will maintain explicit belief states to drive routing, stopping, and budgeting decisions.
2. Value of Information (VOI): Embed Bayesian decision theory to drive act-vs-gather decisions. Trigger additional evidence collection (e.g., running an expensive detector) *only* when the expected reduction in decision loss exceeds the acquisition cost.
3. Human Collaboration: Design uncertainty-driven deferral mechanisms. Calibrated posteriors and utility parameters will route ambiguous, high-entropy alerts to humans, integrating analyst responses back into the belief update loop to reduce workload and improve detection robustness.

Literature Review

- **Agent Orchestration:** Frameworks like the *Model-Control-Policy* (Suggu) dictate the need for a governance layer, while hierarchical multi-agent systems (Alba Torres) validate distributed defense architectures. However, they lack mathematical optimization engines.
- **Bayesian Decision Theory & VOI:** Papamarkou et al. assert agentic AI must make "Bayes-consistent" decisions using a central belief state. Kim et al. successfully applied VOI to alert triage, reducing detection time by 79%, proving the superiority of probabilistic prioritization.
- **Human-AI Collaboration:** Tilbury warns of "alert fatigue" without intelligent filtering. De Nascimento operationalized this by routing high-entropy (uncertain) predictions to analysts, keeping workloads manageable.
- **The Gap:** No unified framework orchestrates diverse detectors with a central belief state while using VOI to dynamically trigger expensive agents or human analysts.

Research Methods: High-Level Architecture

1. LangGraph Orchestration: Governs the system as a stateful directed graph. It holds the shared graph state (current belief, evidence, costs) and implements the VOI Router to conditionally branch to tools, actions, or humans.
2. A2A Protocol: A decentralized communication bus allowing lateral coordination between detector agents to share evidence asynchronously without blocking the main orchestrator graph.
3. LangChain Pipelines: Manages the internal reasoning of specific agents (e.g., log analysis) while keeping them black-boxed from the orchestrator's perspective.
Key Properties: Ensures full state continuity, decoupled coordination, and "feedback as graph re-entry"—where human oversight formally pauses the graph and analyst signals dynamically update the shared belief state.

Research Methods: Bayesian Orchestrator Logic

```
flowchart LR
  subgraph ControlPlane["Control Plane"]
    REG["Agent Registry (YAML)"]
    ORCH["Orchestrator\nLangGraph Flow\nVOI + Decision"]
    POL["Routing Policy\nThresholds"]
  end

  subgraph DataPlane["Data Plane"]
    A2A["A2A HTTP Client"]
    STATE["Shared State Backend\nSQLite"]
  end

  subgraph Agents["Agent Services (Containers)"]
    A["Agent A\nLightweight"]
    B["Agent B\nDeep"]
    D["Agent D\nLLM"]
  end

  REG --> ORCH
  POL --> ORCH
  ORCH --> A2A
  A2A --> A
  A2A --> B
  A2A --> D
  ORCH <--> STATE
```