

MULTIPLAYER GAME STATE SYNCHRONIZATION PROTOCOL (VAP-1)

Violet Ascending Protocol (VAP-1) is a custom UDP-based application-layer protocol designed for real-time multiplayer game state synchronization. It is implemented and evaluated through a simple multiplayer where multiple clients compete to acquire cells on a shared grid. The protocol prioritizes low latency, bandwidth efficiency, and robustness under packet loss and network delay using delta compression, snapshot buffering, and selective reliability.

BUILD INSTRUCTIONS

Prerequisites

Operating System

Linux (Ubuntu) or WSL2 is required to run the automated network tests using tc netem. The game client itself can run on Linux, Windows, or macOS.

Python

Python version 3.8 or higher is required.

Required Python Libraries

select,socket ,enum,time, json,zlib,numpy as np,psutil

HOW TO RUN TESTS

All experiments are automated using a unified shell script called run_tests.sh. This script starts the server, launches multiple headless clients, applies network impairments using Linux Traffic Control (tc netem), logs metrics, and generates result plots.

Run Full Test Suite

```
sudo ./run_tests.sh all
```

Run Individual Test Scenarios

Baseline

```
sudo ./run_tests.sh baseline
```

Packet Loss (2%)

```
sudo ./run_tests.sh loss2
```

Packet Loss (5%)

```
sudo ./run_tests.sh loss5
```

High Latency (100 ms)

```
sudo ./run_tests.sh delay100
```

Generate Graphs Manually

```
python3 plot_summary.py
```

HOW TO RUN THE GAME

Start the Server

```
python3 server.py
```

Start the Client

```
python3 game.py
```

EXPLANATION OF CHOSEN DESIGN MECHANISMS

Binary Header Structure

VAP-1 uses a compact 24-byte binary header containing sequence numbers, snapshot IDs, and server timestamps to reduce overhead.

Delta Compression

Only changed grid cells are sent per update, reducing bandwidth usage. However if the client is falling behind too much the full grid is sent

Snapshot History Buffer

The server stores recent snapshot deltas to recover from packet loss without retransmissions.

Selective Reliability

Critical events such as cell acquisition are acknowledged and retransmitted if needed.

Fixed Tick Rate

The server runs at 25 Hz to ensure stable bandwidth and low latency.