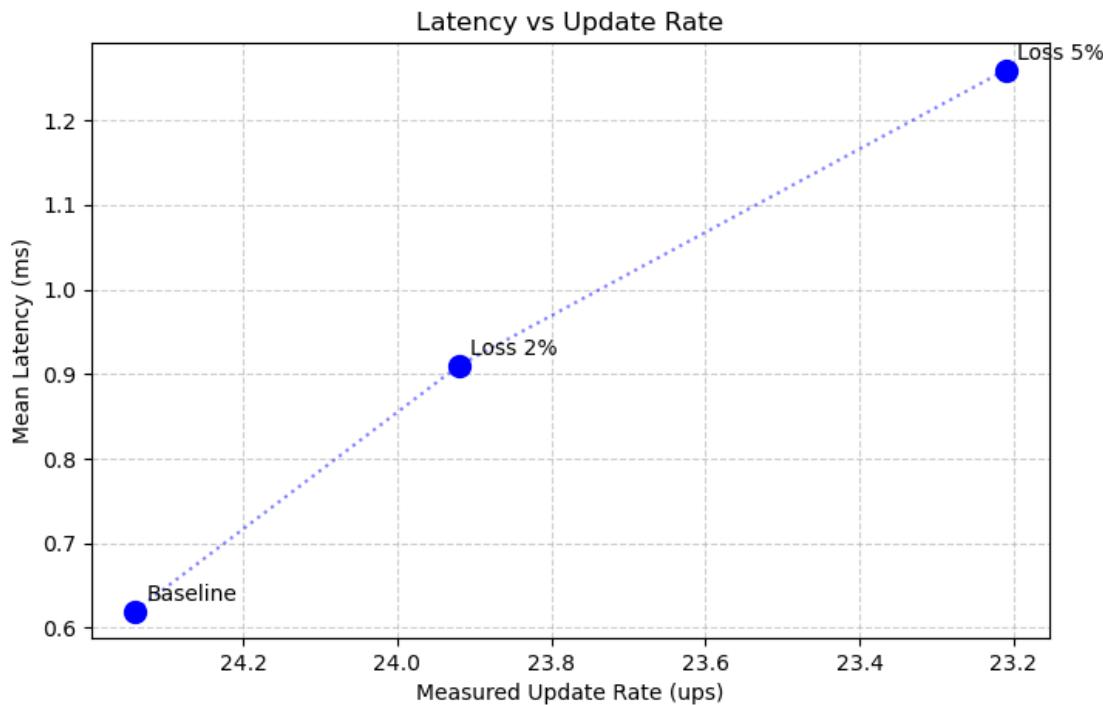
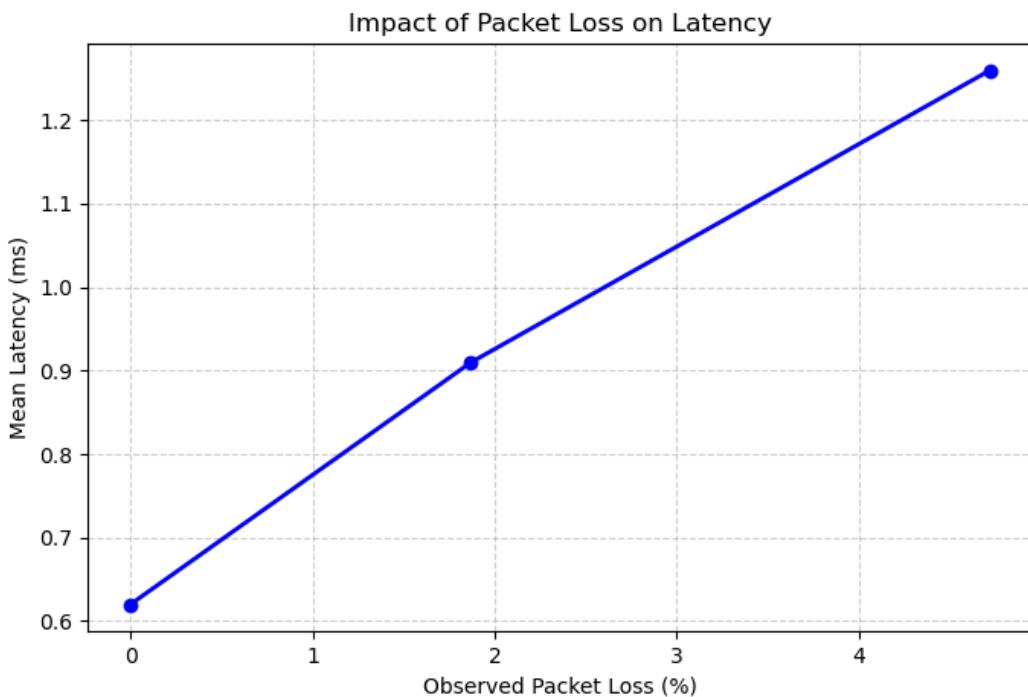


Plots

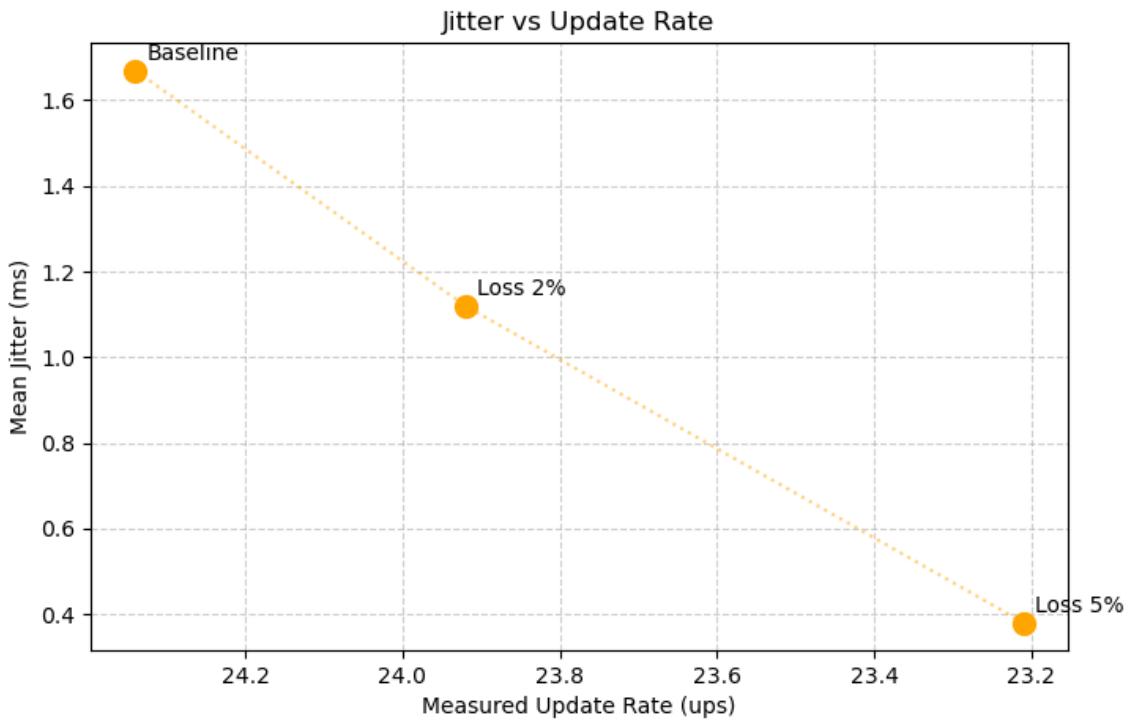
Latency vs Update Rate:



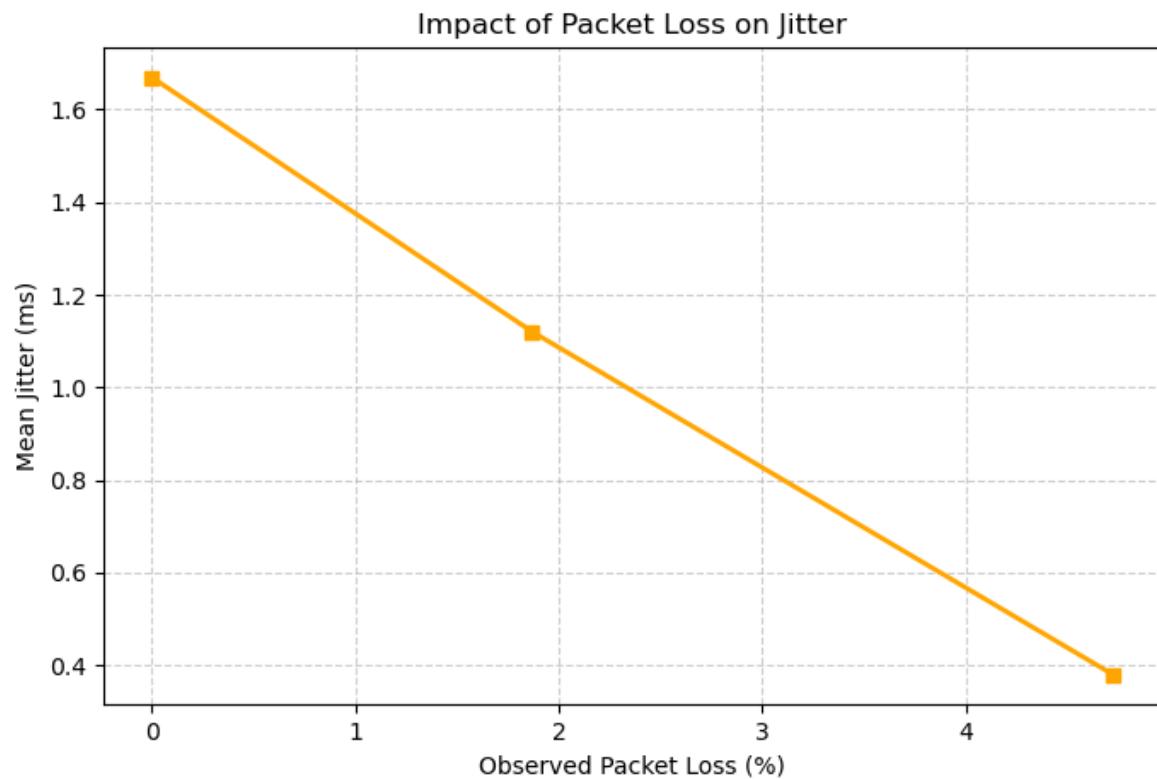
Latency vs Loss:



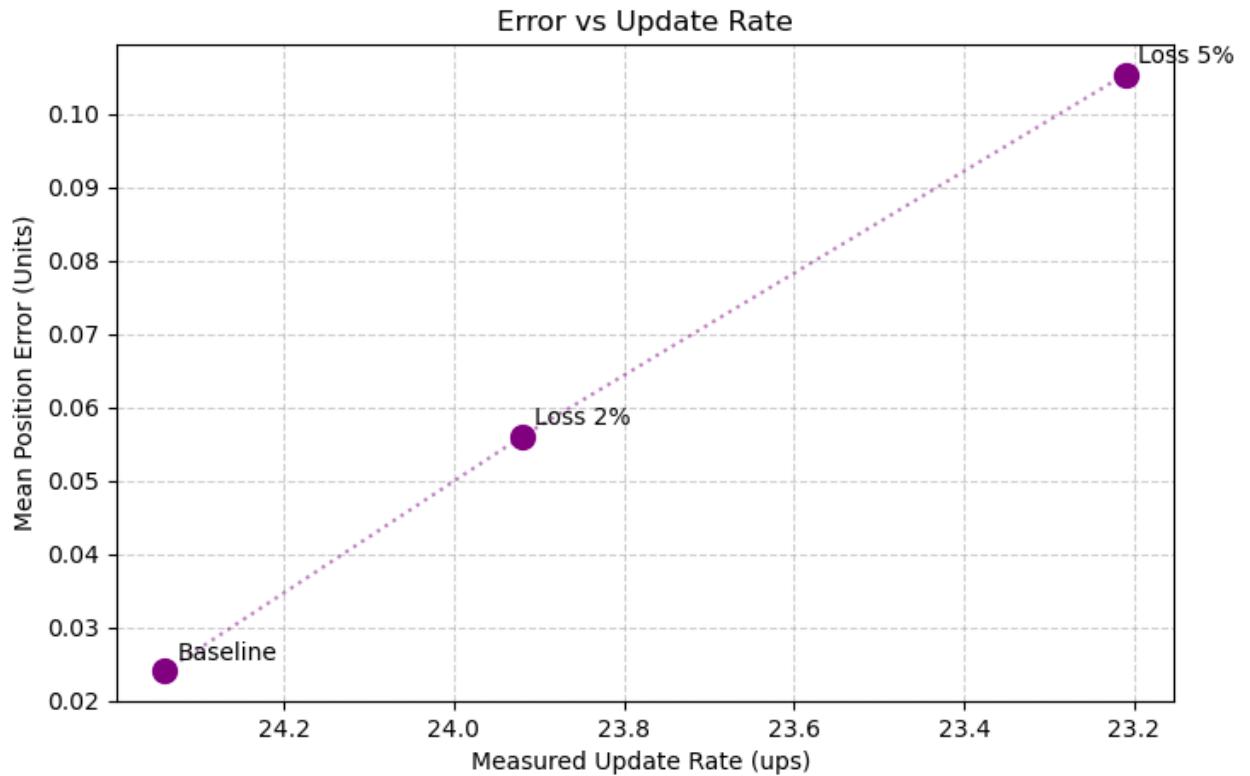
Jitter vs Update Rate:



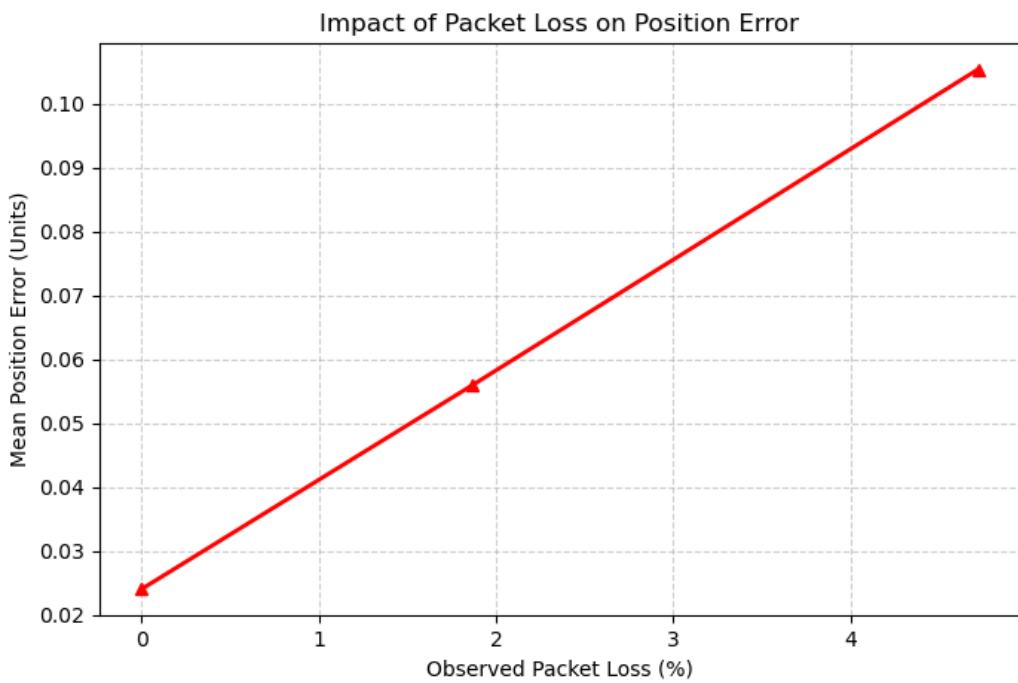
Latency vs Loss:



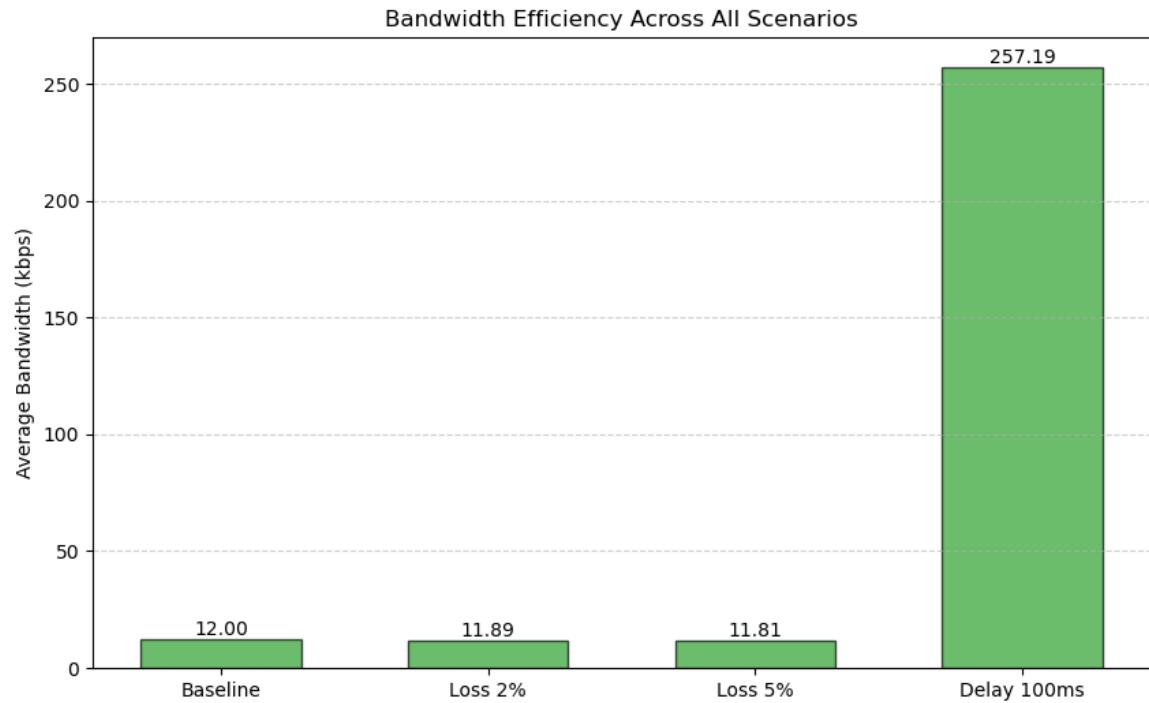
Error vs Update Rate:



Latency vs Loss:



Bandwidth Comparison:



Automated script to run baseline test code:

```
#!/bin/bash

set -e

if [ "$1" == "all" ]; then
    echo "====="
    echo "Running All Tests "
    echo "====="

    echo "Running baseline..."
    sudo bash "$0" "baseline"
```

```

echo "Running loss2..."
sudo bash "$0" "loss2"

echo "Running loss5..."
sudo bash "$0" "loss5"

echo "Running delay100..."
sudo bash "$0" "delay100"

python3 relations_plot.py

echo "Testing Complete"
exit 0
fi

TEST_MODE=$1
if [ -z "$TEST_MODE" ]; then
    echo "Usage: sudo ./run_tests.sh [baseline|loss2|loss5|delay100|all]"
    exit 1
fi

INTERFACE="lo"
RUN_DURATION=130
CLIENTS=4
OUT_DIR="results/${TEST_MODE}/run1"

echo "==== Starting Test: ${TEST_MODE} ===="
echo "Cleaning old results in ${OUT_DIR}"
rm -rf "${OUT_DIR}"
mkdir -p "${OUT_DIR}"

echo "Setting up network conditions on ${INTERFACE}"

sudo tc qdisc del dev ${INTERFACE} root 2>/dev/null || true

case ${TEST_MODE} in
    "baseline")
        echo "    -> Network: No impairment (Baseline)"
        ;;
    "loss2")
        echo "    -> Network: 2% Loss"
        sudo tc qdisc add dev ${INTERFACE} root netem loss 2%
        ;;
    "loss5")
        echo "    -> Network: 5% Loss"
        sudo tc qdisc add dev ${INTERFACE} root netem loss 5%
        ;;
    "delay100")
        echo "    -> Network: 100ms Delay (+/- 10ms jitter)"
        sudo tc qdisc add dev ${INTERFACE} root netem delay 100ms 10ms
        ;;
esac

```

```

        *)
            echo "    -> Unknown mode: ${TEST_MODE}"
            exit 1
        ;;
esac

if command -v tshark &> /dev/null; then
    echo "Starting packet capture..."
    tshark -i ${INTERFACE} -f "udp port 8888" -w "${OUT_DIR}/traffic.pcap" > /dev/null 2>&1 &
    CAPTURE_PID=$!
    sleep 1
else
    echo "[WARN] tshark not found. Skipping pcap."
fi

echo "Launching Server"
python3 -u server.py > "${OUT_DIR}/server_log.txt" 2>&1 &
SERVER_PID=$!
sleep 2

for i in $(seq 1 ${CLIENTS}); do
    echo "Launching Client ${i}"
    python3 -u client.py > "${OUT_DIR}/client${i}_log.txt" 2>&1 &
    CLIENT_PIDS+=($!)
    sleep 0.5
done

echo "Running test for ${RUN_DURATION} seconds"
sleep ${RUN_DURATION}

echo "Stopping processes"
kill ${SERVER_PID} 2>/dev/null || true
kill ${CLIENT_PIDS[@]} 2>/dev/null || true
if [ ! -z "$CAPTURE_PID" ]; then kill $CAPTURE_PID 2>/dev/null || true; fi

sudo chown -R $USER:$USER "${OUT_DIR}"

echo "[INFO] Resetting network rules..."
sudo tc qdisc del dev ${INTERFACE} root 2>/dev/null || true

echo "=====
echo "Collecting metrics from ${OUT_DIR}..."
python3 collect_metrics.py "${OUT_DIR}" "${TEST_MODE}"

echo "Generating plots"
python3 plot_metrics.py "${OUT_DIR}"

echo "Test Complete "

```

```
echo "Logs and Plots saved to: ${OUT_DIR}"
```