Remotely Controlled Car via LTE or Wifi

Christian Prieto cprieto2023@my.fit.edu
Joseph Digafe jdigafe@my.fit.edu
Nicholas Shenk nshenk2023@my.fit.edu
Donoven Nicolas dnicolas2021@my.fit.edu

Marius Silaghi msilaghi@fit.edu

Sep 10, 2025 meet with first client

Goal and Motivation:

Our goal is to provide a unique device that tries to fix issues with their current way of investigating potentially hazardous buildings or generally unsafe areas, efficiently. The motivation was for emergency responders not to have to immediately risk their lives to relay or gather information, where an expendable device could be used instead.

Today, responders rely mainly on manual entry or expensive machinery that is either too costly for smaller departments or too complex to operate efficiently under duress. Existing systems often suffer from limited range, fragile communication links, poor video quality, or a lack of encryption. We believe that our project could remove these drawbacks by offering a quality, lightweight, secure, and resilient teleoperation system. It is a system that can be deployed quickly and function across communication networks (Wifi or LTE), and provide operators with real-time feedback on connection quality.

With this implementation, we hope to lower responder risk and also increase efficiency. Reliable video and controls allow faster decision-making, we believe. This can help teams survey a larger area and adapt when networks fail. Overall, our product will ensure mission-critical information is delivered without risking the lives of personnel unnecessarily.

Approach (Key features of the System)

1. Live FPV Video with Encryption.
Our system will provide operators with a live video feed that is responsive in first-person view (FPV). The video stream is adaptive based on whatever condition the network is in; our video stream will adapt to maintain an end-to-end latency of under 300ms. Something to note about this is that all commands and telemetry are carried over a secure channel crafted and curated by our team. A protocol like this will enforce encryption to a great degree. Not only encryption, but also authentication and replay protection will be handled by the protocol. With this, our operators can trust both the video they see and the commands they choose to send.

2. Unified Operator Console (Laptop and Desktop UI)

All control and feedback for the car will be handled by a dedicated Windows application. The console will not only display live video, but also live video stats: latency, jitter, packet loss, etc. Operators will be able to drive using their preferred input device. Our supported input devices are: gamepad, steering wheel, or keyboard. Also, safety is built into the design. The control will have a panic stop button that lets the operators halt the vehicle instantly. Our integration's goal is to provide operators with a clear and reliable control environment.

3. Connection Resilience and Failover
You never have to worry about losing control during an operation. If your network connection drops, our car will automatically attempt to re-establish the connection while you keep your focus on the task. Video and controls adjust smoothly to whatever bandwidth is available, which makes this car super adaptive. If communication really cannot be restored, the car stops on its own. Therefore, instead of worrying about the link, the operator can stay focused on the environment you are investigating.

4. Flexible Wi-Fi and LTE Operation
The operator can choose how to connect: Wi-Fi for high-speed local use. Or an operator can use LTE for when they need to reach further. If you move out of Wi-fi range, the system quietly switches to LTE in the background. This means there are no settings, no interruptions, or no panic will be involved. A built-in delay meter shows you exactly how responsive the system feel, so you will always and definitely know what to expect. This flexibility gives you confidence that whatever the mission is, the car will stay online and quite ready!

Novel Features / Functionalities:

Custom Secure Protocol: Instead of relying solely on libraries, the team will actually implement its own channel in C++ (handshake, key schedule, authenticated encryption), using published standards and test vectors. This educational contribution will demonstrate a learned deep understanding of security prtocols.

Cross-Network Failover: The system provides automatic fallback between Wi-Fi and LTE networks, which is uncommon in consumer FPV solutions.

Operator Feedback: Real-time delay and connection statistics are surfaced in the UI so users can adapt strategies on the fly.

Algorithms and Tools

Languages: C++ (core system, networking, crypto, UI), Python (scripting/testing).
Networking: Unix/Windows socket APIs, UDP packetization.
Cryptography: Custom implementation in C++ of secure handshake, HKDF key schedule, and AEAD (ChaCha20-Poly1305) with replay protection. ASK advisor about this!!!
Hardware Libraries: libCamera (Raspberry Pi), SIM7600X cellular library, ESP32 libraries for motor/serial interfaces.

UI: Windows API (C++) or Java Script???
Video Encoding: C++ API bindings for hardware/software encoding.

Technical Challenges

1. **Low-Latency Video:** We must integrate video capture, encoding, and UDP streaming with minimal buffering across unpredictable networks.

2. **Cryptographic Implementation:** Implementing AEAD and secure handshakes correctly and securely in C++ requires careful use of test vectors and protocol design.

3. **Cross-Platform Networking:** Supporting both Unix (car side) and Windows (operator side) with low-level socket APIs introduces portability and debugging challenges.

Milestone 1 – Tooling & Core Demos (Sep 29)

- **Video first:** libcamera capture on the car with **adjustable resolution & framerate**; render on laptop.

- **Networking demo:** UDP send/receive on Windows↔ESP32 using raw sockets; timestamp & measure one-way latency.

- **Collab & docs:** Repo, branching, CI; initial Requirements, Design, Test Plan.
  Outcome: End-to-end **unencrypted (or temporarily encrypted)** video+control proof-of-concept with basic latency metrics. Encryption approach **not finalized**—queued for advisor review.

Milestone 2 – Secure Channel & Control Integration (Oct 27)

- **Cryptography decision point (with advisor):** Choose one:

  - **Option A (custom)**: Handshake + HKDF key schedule + **AEAD** (e.g., ChaCha20-Poly1305) + replay window.

  - **Option B (standards-based)**: **DTLS/SRTP** for media + DTLS/AEAD for control.

  - **Option C (library tunnel)**: **WireGuard-style** or TLS 1.3 tunnel, then your own framing on top.

  - **Option D:** Advisor sole recommendation.
- **Implement chosen path** and document limits (key lifetimes, nonce rules, rekey policy).

- **Control loop:** Gamepad/wheel → 50–100 Hz commands → dead-man stop on silence.
- **UI foundation:** Laptop app shows live video and **basic** stats (latency, bitrate).
  Outcome: Secure **alpha**: operator can drive with live video; crypto path chosen + working.

Milestone 3 – Reliability & Demo (Nov 24)

- **Resilience:** Auto-reconnect on IP change; Wi-Fi→LTE fallback (manual port-forward is fine).
- **Adaptation:** Adjust video bitrate and control rate based on simple link health signals (loss/jitter).
- **Telemetry UI:** Show latency, jitter, loss, and (if Option A) key epoch/replay-drops.
- **Fall demo:** Integrated, secure, low-latency drive session; finalize Fall docs.
  Outcome: Robust demo ready for Spring polish (e.g., full NAT traversal, nicer UI, broader testing).

  Explanation:

  The Fall semester milestones focus on establishing the project foundation: selecting tools for networking, cryptography, and video; building "hello world" demos; and resolving integration challenges such as UDP sockets, AEAD encryption tests, and video capture pipelines. By the end of the Fall, the team will have a working operator interface with live video rendering, a secure channel design (with the advisor's guidance on the final cryptography choice), and basic command/control functionality. In the Spring semester, the project will transition from prototyping and initial integration to advanced development and refinement. This includes implementing resilience features such as seamless Wi-Fi/LTE failover, reconnection logic, and real-time performance monitoring (latency, jitter, and packet loss). The Spring will also emphasize system optimization, end-to-end testing, and preparing a polished demonstration that integrates networking, security, control, and video into a robust, deployable solution.

Task Matrix for Milestone 1

Task	Nicholas Shenk	Donovan Nicolas	Christian Prieto	Joseph Digafe
Compare/select technical tools (networking, crypto, video)		✔	✔	
Hello world demos (UDP sockets, test, libCamera capture, UI harness)	✔ (UDP ESP 32)	✔ (UDP Laptop)	✔ (Video)	✔ (UI)
Resolve technical challenges (sockets, crypto coding, encoding API, cross-platform)	✔	✔	✔	✔
Compare/select collaboration tools (GitHub, Docs, Slack/Discord, calendar)	✔	✔		✔
Requirement Doc	30%	30%	20%	20%
Design Doc	25%	25%	25%	25%
Test Plan	25%	25%	25%	25%

Approval from Faculty Advisor

- "I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."
- Signature: ______________________________ Date: ________