Plan 1

Phase 1: Simulation and Initial Setup (Months 1-2)

Router Setup:

Set up the Router Development Environment: Prepare the router for development by setting up the necessary tools and ensuring you can access and modify its firmware.

If you the router gives you a possibility to develop traffic filtering/simulation, Connection Quality Monitoring Algorithm will be developed in the router

If not, the connectivity solution will be developed in the machine where Exo Desktop application is installed

Simulation Environment Setup:

Set up the simulation environment for testing the communication between the router and the EXO Desktop App. The simulation environment can be used to test the algorithm under different network conditions.

Algorithm Design and Initial Implementation:

Design the Connection Quality Monitoring Algorithm: Develop the core algorithm for monitoring connection quality, using metrics like signal strength, packet loss, and latency. The algorithm should output a single metric, score, or state that represents connection quality

Begin Implementing the Algorithm: Start implementing the algorithm within the router's firmware, focusing on the essential logic and data collection.

Initial Communication Protocol:

Design the initial communication protocol to send connection quality data from the router to a simulated EXO Desktop App. This could involve a simple REST API, MQTT, or other network protocol.

Phase 2: Local LLM Setup and Testing (Months 3-4)

Router Integration with Local Setup:

Implement Communication with the EXO Desktop App: Develop the necessary code in the EXO Desktop App to receive connection quality data from the router. This is critical to allow switching between cloud and local modes, based on the quality of the connection.

Integrate Initial Switching Logic: Implement basic logic in the EXO Desktop App to use the connection quality data to decide whether to use the local LLM.

Begin testing the router connection in a real environment with the local LLM setup.

Local LLM Setup and Testing:

Set up the local LLM on the MAC mini, ensuring that it can be accessed from the EXO Desktop App.

Test the local LLM setup on the MAC mini using the EXO Desktop App, and monitor the performance of the LLM running on your hardware1....

Testing and Benchmarking

Carry out performance tests in the local setup to determine the baseline performance when using local models, and check the stability of the setup using the router.

Use the information to fine-tune and improve the setup.

Phase 3: Cloud LLM Integration and Testing (Months 5-6)

Advanced Router Functionality:

Refine Connection Quality Algorithm: Fine-tune the algorithm based on real-world testing, ensuring it accurately reflects connection quality.

Optimize Communication: Optimize the communication protocol between the router and the EXO Desktop App to minimize latency and resource usage.

Test the system with the router connected via the cellular network.

Cloud LLM Integration:

Integrate with a cloud-based LLM system and ensure that it can be accessed via the EXO Desktop App.

Ensure that the EXO Desktop App can switch between the cloud-based and local LLMs.

Testing and Validation:

Extensive Testing: Perform extensive tests with different devices, under varying network conditions to make sure the system is robust.

Test the ability of the EXO Desktop App to switch between the local and cloud-based LLMs, and monitor how long it takes to switch.

System Refinement: Debug, fix bugs, and implement the necessary fixes based on your tests, and make sure the switching is seamless for the user5....

Documentation and Deployment

Documentation: Create comprehensive documentation for the connection quality monitoring algorithm, its integration with the EXO Desktop App, and the router setup [from conversation history].

Deployment: Prepare the necessary scripts and guides for a smooth deployment process and make it accessible to the community12.

Community Engagement: Share your results, code and documentation with the EXO community via the Discord channels and github repo12....

Key Development Areas (Recurring)

API Access: Verify whether the router offers an API to access connection status and quality metrics programmatically.

Communication Protocol: The router should be able to send the connection quality to the EXO Desktop App using a well-defined protocol.

Switching Logic: The EXO Desktop App needs to make decisions about when to switch between cloud and local LLMs based on connection quality.

Testing and Benchmarking: Rigorous testing of this system is required to ensure the system performs as required in different conditions. Use the benchmark suite to check performance and report results1....

Open Source: Share the source code of all the changes with the community. This includes the modifications to the EXO Desktop App and the router software12.

Expected Outcomes

By the end of the six-month period, you should have:

A functional EXO Desktop App running on a system connected via a 4G/5G router.

A router monitoring the quality of the internet connection using your algorithm.

The EXO Desktop App should be able to switch between cloud and local LLMs depending on the connection quality measured by the algorithm on the router.

Comprehensive documentation and the code available for the EXO community to use and further develop.

This updated work plan integrates the router development into the three phases, ensuring that the network component is developed alongside the core functionalities of the EXO system.

Plan 2.

Phase 1: Simulation and Initial Setup (Months 1-2)

Understand the EXO Architecture: Begin by reviewing the provided sources to fully grasp how EXO manages the interaction between cloud and local LLMs. Key concepts include:

Automatic switching: The system automatically switches between cloud and local LLMs based on network availability.

EXO Desktop App: This app is central to managing local LLM processing and network monitoring.

Local LLM: A backup LLM that runs on your device for use when a network is unavailable.

Data Synchronization: The need to synchronize data triggers a mode switch when it is no longer possible.

Set Up EXO Gym: Use EXO Gym to simulate the distributed training environment and test the switching mechanism. This allows you to experiment with different scenarios without needing actual distributed hardware1.

Use the diloco-sim simulator to test how the DiLoCo algorithm works, which is used to train models on low-bandwidth distributed networks.

Experiment with different configurations of network connectivity to simulate a switch from connected to disconnected modes1.

Use provided examples like training a CNN to understand how to use the simulator.

You may want to modify the existing code for the simulation environment to explicitly test the triggers related to the loss of network connection and the need to synchronize data4.

Develop a Basic Application: Create a simple application that can use an LLM for a basic task (e.g., answering questions, summarizing text). This will be the application that will be switched between cloud and local LLMs.

Implement Switching Logic: Within your application, emulate the EXO Desktop App’s behaviour

Monitor network connectivity. Implement a basic network monitoring system within your application that checks for network connection to the cloud.

Simulate data synchronization. Simulate the requirement for data synchronization for a mode change, where a mode change is triggered when the sync is no longer available due to loss of network [from conversation history].

Trigger mode switching. When the simulation detects loss of network, trigger the switch from the cloud LLM to the local LLM, and vice versa.

Phase 2: Local LLM Setup and Testing (Months 3-4)

Set up Local Environment: Prepare the local environment with the required software and hardware.

Install the EXO Desktop App on a Mac, as this is the central component for managing local LLM processing.

Ensure you have the necessary hardware, such as a Mac Mini, that is compatible with the EXO software6.

Integrate Local LLM: Configure the EXO Desktop App to use a local LLM:

The EXO Desktop App uses models like Qwen2.5 with function calling capabilities, running locally for privacy and reliability7.

The app already has integrations for speech recognition using WhisperKit implementation of OpenAI's whisper-v3-large model7.

Ensure that the local LLM has the capability to perform basic text-based tasks to match the application capabilities you built in the first two months of the project7.

Initial Local Testing: Test the basic application that you developed in Phase 1 using the local LLM set up in the previous step and ensure the application can be used when the network connection is disabled.

Test the automatic switching mechanism on your local set up by simulating a connection loss, and confirming that your application correctly switches to the locally hosted LLM [from conversation history].

Data Sync and Switching Testing: Test the data synchronization triggers the automatic switch between cloud and local mode, and also perform data sync when the connection is re-established and the application switches from the local LLM to the cloud LLM.

Phase 3: Cloud LLM Integration and Testing (Months 5-6)

Cloud LLM Setup: Configure access to a cloud-based LLM service.

Ensure your application can connect to the cloud LLM.

Ensure your application can send and receive text to and from the cloud LLM service.

Cloud Integration: Integrate the cloud LLM with your application, so that when a network is available, the system uses it for the application’s LLM needs:

In connected mode, the system should send requests to the cloud-hosted LLM, and receive responses back through the network [from conversation history].

Switching Logic Implementation: Ensure the application incorporates the mechanisms described by the sources, with the EXO Desktop App as a central element:

The system should automatically detect network availability.

The system should manage the transition between cloud-based and local LLMs, where users don't need to manually change settings [from conversation history].

The system should incorporate data synchronization, which can trigger the mode change [from conversation history].

The application should use the local LLM when disconnected from the cloud [from conversation history].

End-to-End Testing: Conduct thorough testing of the entire system by simulating various network connectivity scenarios to confirm seamless switching:

Test switching from cloud to local by disconnecting the network connection, and make sure the system correctly switches to the local LLM [from conversation history].

Test switching from local to cloud by reconnecting the network and confirming the system switches back to the cloud LLM [from conversation history].

Test data synchronization. Test that data is synchronized correctly when switching between local and cloud modes.

Test performance: Confirm that the application performance is acceptable both when using cloud LLM, and when using local LLM.

Final Demonstration: Prepare a live demonstration showcasing seamless switching between the cloud and local LLMs, highlighting the automatic nature of the switch and the consistent functionality of your application in both modes.

By following these steps, you can create a demonstration of the seamless switching between cloud and local LLMs, leveraging the key principles of the EXO system.