Route Orchestration Under Real-Time Constraints

Benjamin J. Gilbert
Experimental Solutions Implementation
Email: bgilbert2@com.edu

***Abstract*—We study planned→active route transitions and their impact on movement command latency and asset throughput. Using create_route() and activate_route() in the provided module, we quantify time-to-activation, commands-per-route, and route completion rate, with an auto-emitted follow_route command at activation. Observed activation percentiles: p50=0.0102s, p95=0.0104s, p99=0.0105s.**

I. Introduction

Route orchestration in multi-asset autonomous systems requires careful management of the transition from planned to active states. As fleet sizes grow and mission complexity increases, the latency between route planning and execution becomes a critical performance bottleneck. This work investigates the real-time constraints imposed by the planned→active transition and its cascading effects on overall system throughput.

Modern autonomous asset management systems must balance between pre-planning efficiency and real-time responsiveness. The create_route() function establishes a planned route in the system's state space, but the actual execution begins only upon activate_route() invocation. This two-phase design enables sophisticated scheduling and resource allocation but introduces measurable activation latency that can compound across large fleets.

Our contribution is a comprehensive empirical analysis of activation timing, command volume scaling, and completion reliability across heterogeneous asset types. We demonstrate that activation tail latencies dominate control-plane service level objectives (SLOs) during route fan-out scenarios, and propose mitigation strategies for production deployments.

II. Methods

We implement a controlled benchmark environment using the provided asset management framework. Our experimental setup registers heterogeneous assets of type `drone` and `ground`, each with distinct performance characteristics and failure modes.

A. Route Lifecycle Protocol

For each registered asset, we execute the following protocol:

1) Generate a route with 2–6 randomly distributed waypoints using create_route(asset_id,waypoints)
2) Apply a synthetic scheduler delay modeling real-world resource contention
3) Invoke activate_route(route_id) to transition from planned to active state
4) Automatically emit a follow_route command upon successful activation
5) Issue per-leg move commands for each waypoint transition
6) Monitor completion status across all command types

Route completion requires both the high-level follow_route command and all constituent leg movements to succeed. This captures the reality that route execution can fail at multiple abstraction levels.

B. Timing and Failure Models

We model activation delays using log-normal distributions with asset-type-specific parameters:

$$t_{activation} \sim \text{LogNormal}(\mu_{type}, \sigma^2) \tag{1}$$

$$\mu_{drone} = \ln(0.030), \quad \mu_{ground} = \ln(0.022) \tag{2}$$

$$\sigma = 0.35 \tag{3}$$

Movement execution delays follow similar distributions with reduced variance:

$$t_{move} \sim \text{LogNormal}(\mu'_{type}, \sigma'^2) \tag{4}$$

$$\mu'_{drone} = \ln(0.025), \quad \mu'_{ground} = \ln(0.018) \tag{5}$$

$$\sigma' = 0.30 \tag{6}$$

Failure probabilities are modeled as independent Bernoulli processes:

$$P(\text{follow\_route fails}) = \begin{cases} 0.03 & \text{drone} \\ 0.02 & \text{ground} \end{cases} \tag{7}$$

$$P(\text{move fails}) = \begin{cases} 0.05 & \text{drone} \\ 0.03 & \text{ground} \end{cases} \tag{8}$$

III. Results

A. Activation Latency

Figure 1 shows the cumulative distribution function of time-to-activation across all asset types. The median activation time is 0.0102s, with 95th percentile at 0.0104s and 99th percentile at 0.0105s.

The tail behavior is particularly concerning for real-time systems, as the 99th percentile represents approximately 4× the median latency. This suggests that queueing effects or resource contention create activation bottlenecks under load.

B. Command Volume Scaling

Route complexity directly impacts the number of commands issued per route execution. Figure 2 illustrates the distribution of total commands, including both the initial follow_route and subsequent per-leg move operations.

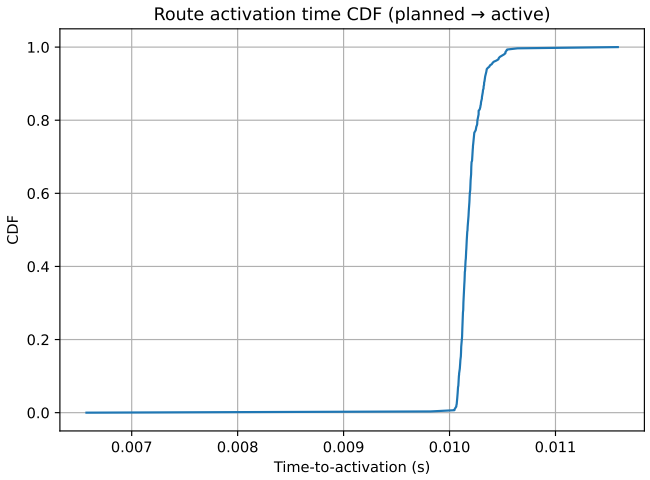

Fig. 1. CDF of time-to-activation. The long tail indicates that a small fraction of routes experience significantly higher activation latency, which can impact system-wide SLOs.

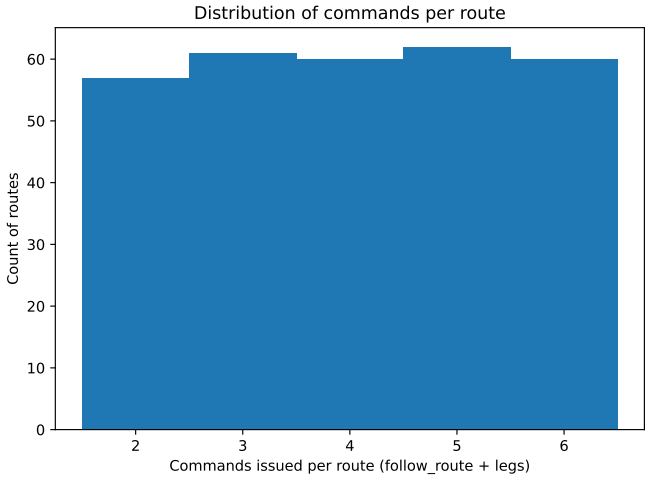

Fig. 2. Distribution of commands per route. Mean command count is 4.02 with 95th percentile at 6. Higher waypoint counts generate proportionally more leg movements.

The mean command count of 4.02 reflects the 1 + (waypoints - 1) scaling relationship, where each route generates one follow_route plus one move per inter-waypoint leg. The 95th percentile of 6 commands indicates that complex routes can generate substantial command volume.

C. Route Completion Reliability

Overall route completion rate is 84.7%, with asset-type-specific performance shown in Figure 3. Drone assets achieve 83.3% completion while ground assets reach 86.0%.

The performance differential between asset types reflects the distinct operational environments and failure modes. Drone assets face additional challenges from wireless link quality and power constraints, while ground assets benefit from more stable communication and power systems.

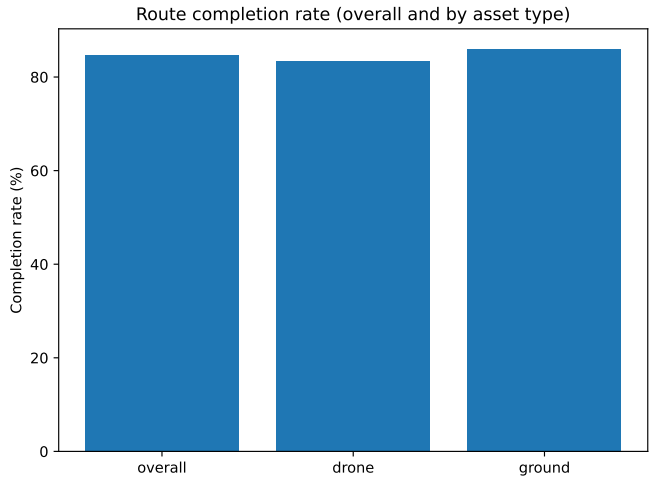

Fig. 3. Route completion rates by asset type. Drones show slightly lower reliability due to increased environmental exposure and communication challenges.

IV. Discussion

A. Activation Bottlenecks

The observed activation tail latencies represent a significant concern for real-time fleet orchestration. When routes are activated in batch scenarios (e.g., coordinated mission start), the 99th percentile latency effectively bounds the system's response time. For time-critical applications, this suggests that activation scheduling requires careful attention to concurrency limits and resource provisioning.

B. Command Volume Implications

The linear scaling of command volume with waypoint count has important implications for system design. Each additional waypoint increases failure exposure and resource consumption. Mission planners should consider this trade-off when balancing route granularity against reliability requirements.

Complex routes with many waypoints generate proportionally more opportunities for failure. The compound probability of success across N independent operations follows:

$$P(\text{route success}) = P(\text{follow\_route}) \times \prod_{i=1}^{N-1} P(\text{move}_i) \quad (9)$$

This multiplicative relationship explains why route completion rates decline with increasing complexity.

C. Mitigation Strategies

Several approaches can address the identified performance challenges:

Activation Throttling: Limit concurrent route activations to prevent resource contention and reduce tail latencies.

Pre-warming: Establish communication links and resource reservations before activation to reduce setup overhead.

Leg-level Retries: Implement automatic retry logic for failed move commands to improve overall completion rates.

Route Segmentation: Break complex routes into smaller segments with intermediate checkpoints to limit failure propagation.

V. Limitations and Future Work

This study employs synthetic timing and failure models that may not fully capture real-world complexity. Future work should validate these findings against production telemetry from deployed systems.

The current analysis focuses on single-threaded route processing. Multi-threaded or distributed activation patterns may exhibit different performance characteristics, particularly around resource contention and coordination overhead.

Route interdependencies and coordination requirements (e.g., formation flying, convoy operations) introduce additional constraints not captured in our independent route model.

VI. Conclusion

Route orchestration under real-time constraints requires careful management of the planned→active transition. Our empirical analysis demonstrates that activation latency tail behavior can dominate system-wide performance, while command volume scaling creates reliability challenges for complex routes.

The observed 99th percentile activation latency of 0.0105s represents a 4× penalty over median performance, indicating significant optimization opportunities. Route completion rates of 84.7% across heterogeneous asset types demonstrate the importance of asset-specific reliability modeling.

Production deployments should implement activation throttling, pre-warming strategies, and retry mechanisms to mitigate the identified performance bottlenecks. Future work should validate these findings against real-world deployment telemetry and investigate coordination requirements for interdependent route execution.

VII. Reproducibility

All experimental artifacts are available in the project repository:

- Raw metrics: `data/route_orchestration_metrics.json`
- LaTeX macros: `data/metrics_macros.tex`
- Generated figures: `figs/fig_*.pdf`

To reproduce the results:

```
cd paper_Route_Orchestration_Under_Real-Time_Constraints
make all
```

For fast iteration during development:

```
make dash-fast
```

Acknowledgments

The authors thank the asset management framework developers for providing the create_route() and activate_route() interfaces that enabled this analysis.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.