

Oren Moreno

ITAI 4370

Professor Tawanda Chiyangwa

December 4, 2025

### Final Reflection Essay

Over the span of this course, telecommunications went from feeling like an invisible background service to something I could reason about structurally and practically. At first, “the network” was mostly an abstract idea: routers, links, and latency were loosely understood ideas, and it was easy to treat the whole system as a black box that delivered packets from point A to point B. Through simulations, debugging, and connecting models to real network behavior, that picture changed. I began to see networks as graphs with nodes, edges, capacities, queues, and policies that can be analyzed, stressed, and optimized.

Working with network simulations was a big part of this shift. Representing routers as nodes in a graph, with explicit neighbors and link capacities, forced a more concrete mental model. When code threw a `KeyError` trying to access a non-existent node, it was not just a Python problem. It highlighted that in real networks, nodes and links can fail or disappear, and robust algorithms must handle changing topologies. Seeing how traffic moved through the graph, where queues formed, and how congestion built up on bottleneck links made concepts like throughput, latency, and packet loss feel less abstract. They became observable outcomes of specific design and load conditions. My understanding of telecommunications also matured into a layered perspective. Even if I was not explicitly coding each layer of the OSI model, I had to be clear about which parts I was implicitly modeling. Assuming reliable links, for example, bakes in a view of what the lower layers are doing. Thinking about 5G and 6G architectures added another dimension. Instead of picturing a centralized core with passive endpoints, I started to see a distributed environment where base stations, edge servers, and user devices can compute, store data, and collaborate. In that view, a telecommunications system is not only about transporting bits, it is also about orchestrating communication and computation across many intelligent agents.

The course also pushed me to apply AI concepts directly to telecommunications problems rather than treat them as separate domains. Simulations became testbeds where AI could influence routing, congestion control, or resource allocation. Even when I did not fully implement learned policies, thinking of routers as agents whose decisions could be optimized by data made it clear where machine learning can add value. Routing choices, queue management, and load balancing are all candidates for adaptive, data driven strategies. Federated learning was a key bridge between AI theory and realistic network constraints. Working with training loops where multiple clients performed local updates and then contributed to a global model made sense in a telecommunications context where data lives at the edge. Issues like client dropout,

variable data sizes, and intermittent connectivity are exactly the challenges that a real 5G or 6G deployment would face. When I encountered an `IndexError` in a `DataLoader`, it reminded me that distributed training pipelines are fragile and that robustness matters even at the experimental stage. Thinking about how a federated system would behave over an actual network helped connect abstract algorithms to practical deployment concerns. Evaluating model performance also took on new meaning in this context. Confusion matrices, sensitivity, and specificity were not just academic metrics. They mapped to real consequences in network tasks, such as misclassifying flows, missing anomalies, or misallocating limited resources. Learning to interpret these metrics carefully, rather than chasing a single accuracy value, is important for any AI that will influence telecommunications infrastructure, where false positives and false negatives can have very different costs.

On the core machine learning side, designing and training neural networks using libraries like PyTorch deepened my intuition for architecture choices. Selecting `Conv2D`, linear, and softmax layers for specific tasks, even in simpler settings, gave me a mental toolkit for how I might approach time series traffic data, classify network events, or detect anomalies. The models I worked with in this course are not production systems, but they made the connection between architecture decisions and data properties much more concrete. Throughout this process, my technical skills grew in noticeable ways. I became more comfortable reading stack traces, diagnosing where and why simulations failed, and distinguishing between superficial bugs and deeper modeling flaws. Libraries like NetworkX, Mesa-style agent frameworks, and PyTorch went from feeling unfamiliar to feeling like tools I could navigate with some confidence. I improved at structuring code around agents, environments, and training loops, which is directly applicable to AI in telecommunications. My systems thinking also improved. I learned to connect low level phenomena like queues on individual links to higher level metrics like end-to-end latency and overall throughput, and then relate those to user experience. Similarly, I started to view AI models as components inside a larger system, not as isolated artifacts. Questions about scalability, robustness, and behavior under stress became part of how I evaluated designs. That mindset is essential when AI is embedded inside critical infrastructure.

Communication and reflection were another area of growth. Writing about simulations, explaining the behavior of models, and documenting design decisions helped me practice translating technical details into clear narrative. Since telecommunications and AI often involve cross-disciplinary stakeholders, the ability to explain what a system is doing and why it behaves as it does is just as important as writing code. There are still clear areas where I want to grow. On the theoretical side, I want a stronger foundation in queuing theory, stochastic processes, and information theory so that I can complement simulation intuition with rigorous analysis. I am also interested in learning more about reinforcement learning for network control, since many telecommunications decisions are naturally sequential and feedback driven. To apply reinforcement learning effectively, I will need a better grasp of algorithms, stability issues, and how to design state and reward spaces that reflect real network goals. I also want to get more

experience with realistic networking tools and datasets. Educational simulations are invaluable, but there is a gap between a controlled model and the messy reality of protocols, wireless propagation, and hardware constraints. Working with simulators like ns-3, real traffic traces, or testbeds would help bridge that gap. On the AI side, I want to deepen my skills in privacy preserving techniques such as differential privacy and secure aggregation, since telecommunications data often involves sensitive user information.

Finally, I see an ongoing need to improve at building end-to-end systems that integrate models, monitoring, logging, and operational safeguards. A model that works in a notebook is only one piece of a telecommunications AI pipeline. Real deployments need continuous evaluation, clear alerting, and mechanisms for human oversight when the AI behaves unexpectedly. Overall, this course transformed telecommunications from a background assumption into a domain that feels concrete, dynamic, and closely tied to AI. By building and debugging simulations, experimenting with federated learning, and evaluating model performance in a network context, I developed a more mature understanding of both fields and how they intersect. At the same time, I have a clearer view of the skills and knowledge I still need to develop, which makes this intersection of AI and telecommunications a compelling direction for future work.