

Wait a Second! How Delays Disrupt Optimal Policies

Sarah Dumnich
Saint Vincent College
sarah.dumnich@stvincent.edu

Britton Wolfe
Grove City College
wolfebd@gcc.edu

Abstract

Communication delays can lead to inconsistent views of the world among different agents. We prove that an optimal policy in a system with instant communication can be suboptimal when the same system exhibits communication delay.

Overview and Related Work

In a distributed multi-agent system, instantaneous communication is impossible. Instead, each agent has outdated information from each of the other agents. This results in each agent having slightly different versions of the state of the world. We model agents' different views of the world as distinct, possibly conflicting, state vectors, rather than noisy observations of a single state vector (Dumnich, Birmingham, and Wolfe 2023). This formulation is well-suited to virtual environments such as video games, where there is not a central authority or single "true" state, but each agent's view of the world is equally valid. If communication delay is ignored when an AI bot is learning to play the game, the learned policy may prove suboptimal when later acting in a real distributed system, under delay. We analyze two games in which the presence of delay will change the optimal policy, a simple game where we prove there is a different optimal policy and a complex game we analyze empirically.

Our empirical analysis examines a learning agent in a multi-agent system with continuous states, continuous observations, asynchronous actions, no synchronized clock, and stochastic communication delays between agents that prevent agents' views of the world from being completely synchronized. Previous work makes additional assumptions about the system that our work relaxes. For example, Spaan et al. (2008) require that agents have a synchronized clock. Messias et al. (2013) did not require synchronous actions, but their agents are given a model of the system rather than learning it from exploring the world like our agent does.

The amount of delay for agents' communications can have a significant impact on how "out-of-sync" those views of the world are. This work explores how the likelihood and amount of delay impacts the quality of the agent's policy.

Copyright © 2025 by the authors.

This open access article is published under the Creative Commons Attribution-NonCommercial 4.0 International License.



Figure 1: Example system with shooter, target, and health

Proof of Suboptimality for Simple System

In our game, there are three aspects of state: location of the shooter, location of the target, and the health of the target. The target's health begins at 1.0, and the game ends whenever the target's health is reduced to 0.0. The game world has three locations: one on the left, one in the center, and one on the right. The target and shooter can be in any of these three locations (Figure 1).

The shooter has three possible actions that they can take: move toward the target, shoot the Klobb, and shoot the Golden Gun. The Klobb's effectiveness depends upon the distance between the shooter and target. If they are both in the same location, the Klobb reduces the target's health by 0.75. If they are distance one apart, it reduces the target's health by 0.50. Finally, if they are distance two apart, it reduces the target's health by 0.25. The Golden Gun will reduce the target's health to 0 if the target and shooter are in the same location but will otherwise have no effect. At any given time, the target will either stay still, move to the right, or move to the left, each with equal likelihood. There is a discount factor of 0.9 and reward of 1.0 upon killing the target.

Delayed information will affect game play for the shooter. In this case, the state information about the target's location is being sent to the shooter. We compare the scenarios where the information is transmitted instantaneously and where there is a delay of one time step (i.e., the shooter sees where the target was before the target's most recent action). With delayed communication, when the shooter "sees" the target directly in front of himself, it is possible that the target has already moved out of the way. Since the shooter has the choice of whether or not to use actions that require precision, the presence of delay affects the policy.

There are states in which the optimal policy under no delay is different from the optimal policy under delay of one time step. One of the most notable is when the shooter and

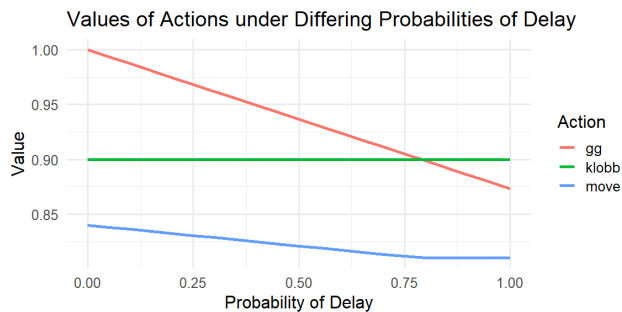


Figure 2: Action values vs. delay probability. The optimal action is different depending on the probability of delay.

target are both in the center of the game (distance 0 apart) and the target has perfect health. Here, when acting under no delay, the unique optimal action is to use the Golden Gun. This is an obvious optimal action as it would be game ending. However, when acting under delay, this is no longer optimal since there is only a 1/3 chance that the target is actually directly in front of the shooter. In this case, using the Klobb is the unique optimal action. What makes this case stand out is that the only reason the Golden Gun is optimal when acting under no delay is that you can guarantee a certain level of precision. In the case of delayed information, it is no longer an effective action, and the agent should instead choose an action that does not require precision, even if it may be less effective.

If not all messages are delayed, as the probability of delay increases, the value of shooting the Golden Gun decreases linearly. It drops below the value of shooting the Klobb when there is 80% chance of delay, yielding a different optimal policy (Figure 2).

Empirical Analysis of Complex System

While the previous system allows us to analytically compute the optimal policy, it is a very simple system. We also analyzed a more complex system: a 3-player first-person shooter simulation, with one learning agent and two agents with hand-coded policies. Each agent can step forward, left, right, rotate left or right, or attack. The learning agent has an additional action available to it, a special attack that has a longer range but requires more precision (i.e., a narrower window to hit the opponent). When any agent is successfully attacked, the episode ends and the game restarts. A successful attack receives a reward, while a missed attack is penalized, and each other action has a small penalty.

This environment has several notable differences from the previous environment: three agents instead of just two; continuous state space instead of discrete; asynchronous actions instead of synchronized time steps; a dispute resolution mechanism via message passing and majority voting, rather than trusting one agent’s view of the world; and stochastic delay times for the messages.

This environment has several notable differences from the previous environment: multiple agents instead of just two; continuous state space instead of discrete; asynchronous ac-

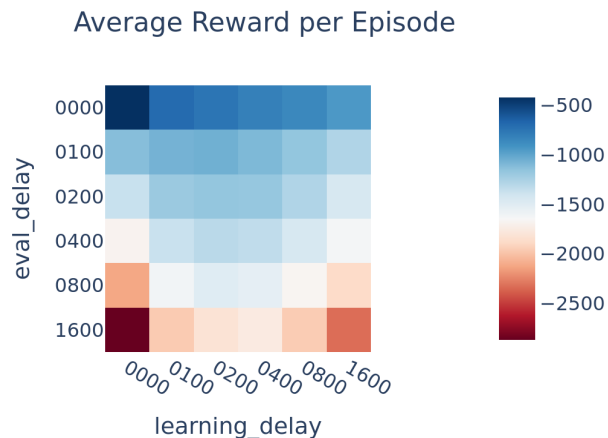


Figure 3: Average reward per episode when evaluating policies learned under different amounts of delay. Darker blues are better.

tions instead of synchronized time steps; a dispute resolution mechanism via message passing and majority voting, rather than trusting one central agent’s view of the world; and stochastic delay times for the messages.

Our learning agent used Q-learning to find its policy (Sutton and Barto 1998). The state variables are the location and orientation of the closest foe, expressed in polar coordinates relative to the learning agent. The agent initialized all Q-values to 0.0 and used an ϵ -greedy exploration strategy ($\epsilon = 0.5$) with learning rate 0.2.

We learned a different policy for several different amounts of delay, with mean delay 0, 100, 200, 400, 800, and 1600 milliseconds. Each message’s delay was sampled from a Weibull distribution with shape 1.5. Each policy was then evaluated under each delay condition, to evaluate how well policies learned under one condition would transfer to other conditions, measured using the average reward per episode (Figure 3). For 0-delay, instantaneous communication, the policy learned under no delay performs the best. For mean delays between 100 and 800, the policy learned under mean-200 delay performs the best, although the policy learned under mean-100 delay is close to the best for network delays of 100 and 200 (Figure 3). The policy learned under no delay makes heavy use of the special attack, which is not optimal in environments with delay. This illustrates that an agent using a policy learned under no delay would be acting suboptimally in a system with delay. The amount of disadvantage varies depending on the precision required for the special attack to count as a hit, from an average disadvantage of 631 when near-perfect precision is required down to 97 when the attack radius was three “steps” of an agent.

Acknowledgments

We are grateful to William Birmingham for the inspiration behind this work.

References

- Dumnich, S., and Birmingham, W. 2020. It's about time: Vector clocks and distributed systems. In *Proceedings of the Second Joint SIAM/CAIMS Annual Meeting*.
- Dumnich, S.; Birmingham, W.; and Wolfe, B. 2023. Embracing the lag: Real-time challenges in multi-agent systems. In *Proceedings of the 2023 AAAI Fall Symposium Series*.
- Messias, J.; Spaan, M. T. J.; and Lima, P. U. 2013. Asynchronous execution in multiagent pomdps: Reasoning over partially-observable events. In *Proceedings of the Eighth Annual Workshop on Multiagent Sequential Decision-Making Under Uncertainty (MSDM-2013)*.
- Spaan, M. T. J.; Oliehoek, F. A.; and Vlassis, N. 2008. Multiagent planning under uncertainty with stochastic communication delays. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling*.
- Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. MIT Press.