*By* Mark Claypool *and*
Kajal Claypool

# LATENCY AND PLAYER ACTIONS IN ONLINE GAMES

*Latency determines not only how players experience online gameplay but also how to design the games to mitigate its effects and meet player expectations.*

The growth and penetration of broadband access networks to the home is fueling the growth of online games played over the Internet. As we write this, it is 5 A.M. local time on a weekday morning, and Gamespy, a source of game server browsing and player forums (www.gamespyarcade.com/), is reporting that more than 250,000 players are playing approximately 75,000 games online worldwide. This proliferation is matched by equivalent growth in the diversity of the games and in available network support.

The best-effort Internet, with no guarantee of network capacity or packet delivery, is a challenge for the real-time interaction required for most online games. By design, most online games have low bit-rate requirements, sending frequent but

*Illustration by Lisa Haney*

small packets typically well within the capacity of broadband or even dial-up modem connections. Moreover, the effects of packet loss are mitigated by frequent game-state updates and repair techniques. These mitigations result in delayed delivery of packets, often in the form of network latency from player to game server or to other players, as the primary bottleneck in online game performance.

Most online games run on a client-server architecture with a single, authoritative server designed to handle game logic. If the latency between the client and the server is large enough, the responsiveness of the game to a player's action decreases, and the player's performance is likely to degrade.

Latency for an action can be attributed to many network components, including the time needed to transmit and receive the encoded action in an IP packet, the time for the packet to propagate from one link to another, and the time spent waiting in a router queue during network congestion. A number of studies have empirically determined lower bound latencies for different types of networks.

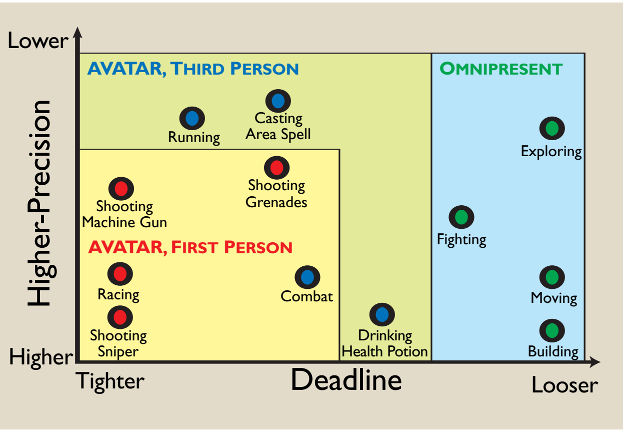

Figure 1. Taxonomy of player actions in the precision-deadline plane, where precision is the accuracy needed to complete the action, and deadline is the time until the final outcome of the action.

Typical LAN latencies, including those in wireless LANs, are small, usually less than 10msec. For game players at home, latencies often depend on the "last-mile" access network. Dial-up modems can add hundreds of msec of latency, while broadband access networks (such as cable and asymmetric digital subscriber lines) typically involve tens of msec of latency [8]. Cable modem latency can, however, vary considerably, with worst-case latencies of more than 100msec. Once on the Internet, latencies are often 50msec within a continent and higher across continents. Overall latencies can be in the hundreds of msec up to more than one second for some Internet connections [7].

Propitiously, not all player interactions are equally sensitive to latency. Some actions (such as shooting a sniper rifle at a moving target) are greatly affected by latency, while others (such as selecting troops and moving them across a battlefield) tend to be less sensitive to latency. To explain this variable effect, we offer our novel categorization of the effects of latency on different player actions based on two salient properties: the precision required to complete the action and the deadline by which the action must be completed. Actions with greater precision and tighter deadlines are sensitive to even modest latencies, while actions with less precision and loose deadlines are more impervious to Internet latencies. The effect of latency on actions can be categorized by the precision and deadline demands of the action, the game's interaction model, and the player's perspective. These classifications are validated by considering player actions in online games, including Warcraft III and Madden NFL, providing a framework for studying and engineering future online games.

## Categorizing Player Actions

Player actions in a game may be categorized along two primary axes: precision and deadline. Precision is the accuracy required to complete an action successfully. Precision is the size of a distant opponent in the scope of a sniper rifle. Deadline is the time required to achieve the final outcome of the action. Deadline is the time to target an opponent with a gun before the opponent moves out of range. The precision and deadline requirements for a player action determine the effects of latency on that action.

Consider a shooting action in which the player targets an opponent moving across the field of view from left to right. With a high-precision weapon, say, a sniper rifle, the player sees the opponent within a tight target circle. When the player aims and shoots, the gun hits any opponent within the circle. With network latency between the player action and the game server recording the action, the opponent is no longer within the circle, resulting in a miss.

However, when the player is shooting a weapon with lower precision (such as a machine gun), the target circle is larger. In this case, the latency between a player action and the game recording that action still allows the opponent to move, but the opponent remains within the target circle, enabling the player to score a hit. This example illustrates our insight, that for a given game action and the greater the precision required, the greater is the effect of latency on player performance.

Consider a movement action in which a player must traverse a suspended beam to reach a treasure. If the beam is straight, then the time to complete the action (the deadline) is large relative to the player-initiated command to move, so additional delays induced by latency do not affect performance much. However, if the beam is twisty, then there are many smaller move commands, each with a tight deadline; even a small delay induced by network latency will significantly impede a player's movement or cause a fall. This example illustrates another insight, that for a given game action and the tighter the deadline, the greater is the effect of latency on player performance.

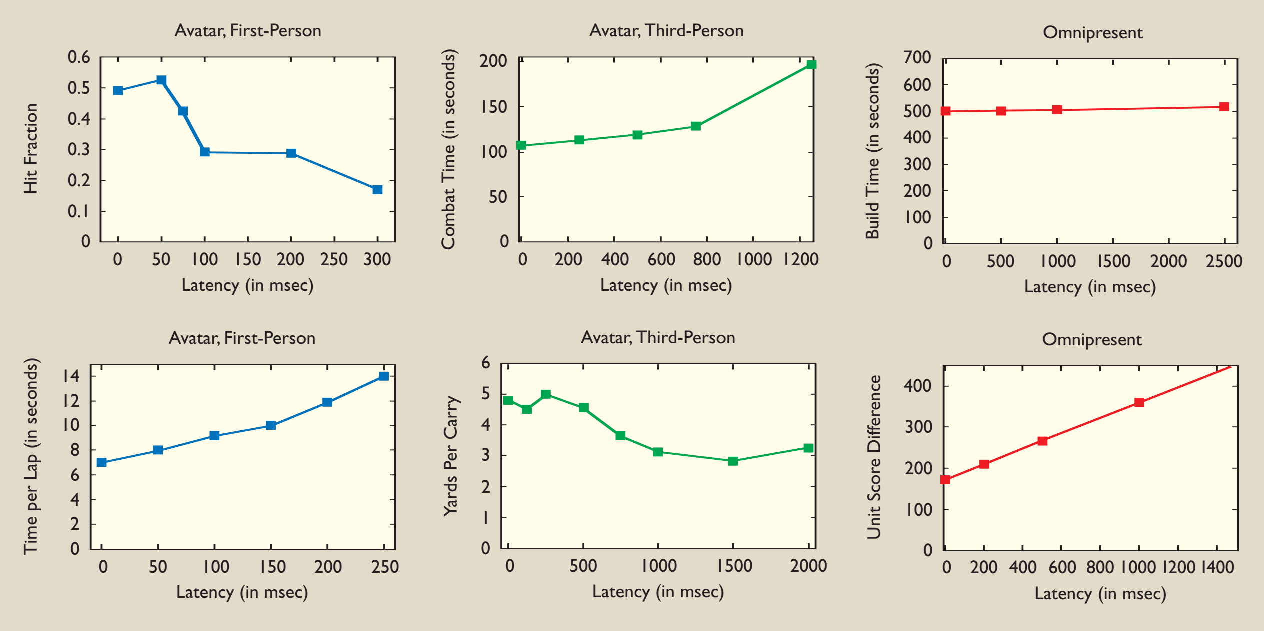

**Figure 2. Player performance versus latency for individual actions.**

Determining the precision and deadline requirements of an action in the precision-deadline plane is key to determining the effect of latency on the player's performance for that particular action. The precision and deadline requirements are determined not only by the action itself but also by the interaction model and by the player's game perspective.

The interaction model defines how a player interacts with the game world and is typically classified as either the avatar or the omnipresent model. In the avatar model, the player interacts with the game through a single representative character, and player actions are defined in terms of commanding it. This avatar, exists at a particular location in the virtual world and can influence only the immediate locality. First-person shooter games, role-playing games, action games, sports games, and racing games are all examples of game genres with an avatar-interaction model. In the omnipresent model, players view and simultaneously influence the entire set of resources under their control. Real-time strategy games and construction and simulation games are genres with an omnipresent interaction model.

The game perspective defines how a player views the game world on a screen. Games with the avatar-interaction model typically have either a first-person perspective where the player sees through the eyes of the avatar or a third-person perspective where the player follows an avatar in the virtual world. The perspective used by games with the omnipresent-interaction model is often variable, giving players an aerial perspective or bird's-eye view of the virtual world while also allowing them to zoom in to a third-person or even a first-person perspective for finer granularity of control over individual resources.

For games with the avatar-interaction model, first-person games tend to have greater precision and tighter deadline requirements than third-person games, since the first-person perspective demands immediate feedback for effective control. By the same token, omnipresent games tend to have looser deadlines and often lower precision requirements than games with an avatar model, as omnipresent games often involve more indirect interaction.

Figure 1 outlines a precision and deadline classification of some player actions. It is based on the action, as well as on the game's interaction model and perspective. The x-axis is the deadline; the y-axis is the imprecision (indicated in the figure as Precision). Shooting with a sniper gun in the avatar model with a first-person perspective reflects high precision and a tight deadline. Running in an avatar model and third-person perspective game typically reflects low precision but a mid-level deadline, while fighting and other actions in an omnipresent-model game typically have looser deadlines and variable precision. In general, the farther away an action is from the origin in the Precision-Deadline plane, the less important is the effect of latency on player performance. Thus, actions in games using an avatar model with a first-person perspective are typically more sensitive to latency than actions in games with an avatar model and a third-person perspective. Actions in games with an omnipresent model are generally less sensitive to latency.

### GAME PERFORMANCE

A number of studies have measured the effect of latency on performance for player actions in games [1–6, 9, 10]. They generally set up a network testbed for the careful control of network latency, typically by having the online game played on a LAN and adding middleware to control network delay. While these studies were not informed by the insights we've provided here, they allow both validation of the classification we present here and illustrate the effects of latency on performance for player actions.

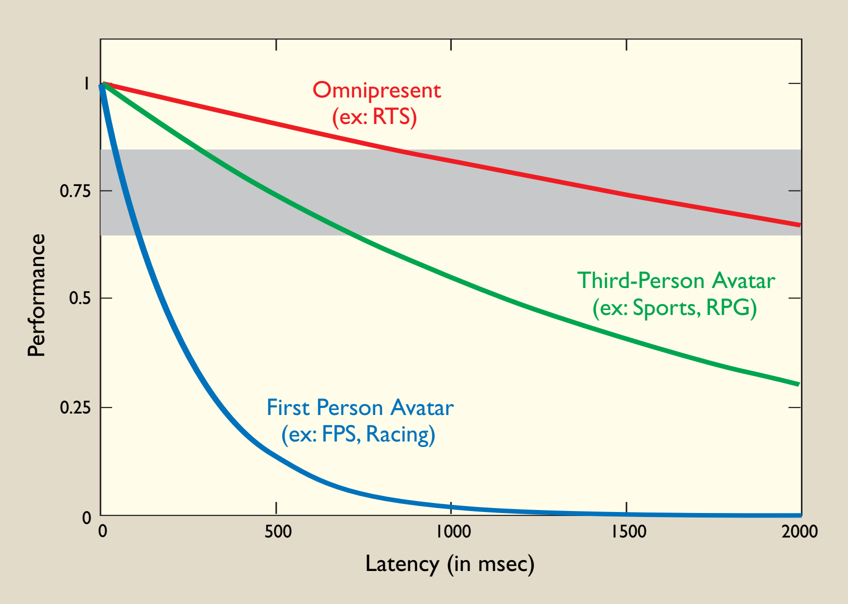

Figure 3. Player performance versus latency for game categories. The horizontal gray band represents the threshold for player tolerance for latency.

Figure 2 includes six graphs of the effects of latency on performance for the different games studied. The x-axes are the amount of latency in msec induced in the experiments; the y-axes are the quantitative measures of performance specific to the particular game studied. For some games, a higher number on the y-axis (such as the accuracy of shooting a weapon) is better. For others, a lower number (such as the time to drive a car around a lap on a race course) is better.

The graphs on the left show the effects of latency on two games with the avatar model and a first-person perspective. The top-left graph depicts the effects of latency on shooting a high-precision gun in Unreal Tournament 2003 [1]. The experiments measured the average hit fraction during two-player battles using high-precision weapons. There is a noticeable overall downward trend in performance as latency increases, with a sharp drop (about 35%) in accuracy at 100msec of latency. The bottom-left graph depicts the effects of latency in a car-racing game [10]. The experiments measured the time to complete a lap around a race track for subjects with varying degrees of driving skill. There is a noticeable upward trend in lap time as latency increases, with a significantly steeper increase above 50msec and again above 150msec.

The graphs in the middle show the effects of latency for two games with the avatar model and a third-person perspective. The top-middle graph depicts the effects of latency on the length of combat in Everquest 2 [5]. These experiments had players do battle on an actual Everquest server over the Internet while inducing a controlled amount of additional latency near the client. With increased latency, the avatar's ability to cause damage decreased, making it take longer to complete a fight. However, the relative decrease in performance from 0msec to 500msec is small, adding only five seconds to about two minutes of combat gameplay. The bottom-middle graph depicts the effects of latency on Madden NFL 2004 [9]. The experiments had players attempt to run the ball, measuring the average number of yards gained per attempt. While there is an overall downward trend in the effects of latency, the experiments found no significant drop-off in performance until after 500msec of latency.

| Model | Perspective | Example Genres | Sensitivity | Thresholds |
|---|---|---|---|---|
| Avatar | First-person | FPS, Racing | High | 100 msec |
| | Third-person | Sports, RPG | Medium | 500 msec |
| Omnipresent | Varies | RTS, Sim | Low | 1,000 msec |

Latency and online games.

The graphs on the right show the effects of latency for two games with the omnipresent model. The top-right graph depicts the effects of latency on the time players need to build everything in the technology tree in Warcraft III [3]. The experiments measured the build time versus latency for all experiment runs, providing a best-fit line for the data. When there was no induced latency, building the technology tree took about eight minutes, while latency values of up to 2.5 seconds increased total build time by 14 seconds.

The bottom-right graph depicts the effects of latency on the unit score difference from combat between two armies in Age of Mythology [3]. The experiments pitted two players with two small, equally matched armies against one another. The unit score difference is the unit score of the player without latency minus the unit score of the player with latency. There is an upward trend in that the score difference increases as latency increases, but the difference in score from no induced latency to one second of induced latency is equivalent to only about one unit, an insignificant amount in the wide-ranging battles typical of most real-time strategy games.

Figure 3 outlines performance versus latency for different classes of online games. The curves are created by normalizing the previously measured performance data, from 0 (worst) to 1 (best), and fitting an exponential curve to the data. Online games with the avatar model of player interaction are more sensitive to latency than games with the omnipresent model. Deeper within the avatar model, games with a first-person perspective are more sensitive to latency than games with a third-person perspective. Within a given game, player actions that are less precise or have looser deadlines tend to shift the curves in the figure to the right, while more precision and tighter deadlines shift the curves left.

Within a given game category (such as first-person shooter and real-time strategy), the relative amounts of different player actions determine the exact location of the curve. For example, a first-person shooter with more movement and less precise shooting may have the blue curve in the figure shifted more to the right and flattened.

The horizontal gray area in the figure is a visual indicator of player tolerance for latency. Gameplay quality is generally acceptable above the gray area and unacceptable below it. The exact latency tolerance threshold depends on the game and to some extent the player's own perception and sense of immersion (hence the gray area). The results summarized in the table of this effect of latency on performance in online games are useful for:

*Game designers.* So they know the latency tolerances of different player actions, helping them apply latency compensation techniques, as needed;

*Network designers.* So they are able to create infrastructures providing quality of service (QoS) for online games and other interactive applications; and

*Game players themselves.* So they are able to make informed choices about their Internet connections and QoS purchases affecting latency and hence gameplay.

Internet latencies can degrade gameplay for all kinds of Internet games, but the degradation is most notable for games with an avatar model, especially those with a first-person perspective (such as Half-Life and Doom). Avatar-model games with a third-person perspective (such as World of Warcraft and Everquest) degrade modestly with latency. Games with an omnipresent model (such a Warcraft and Battle for Middle Earth) are more resilient to the effects of latency. C

## References

1. Beigbeder, T., Coughlan, R., Lusher, C., Plunkett, J., Agu, E., and Claypool, M. The effects of loss and latency on user performance in Unreal Tournament 2003. In *Proceedings of ACM Network and System Support for Games Workshop (NetGames)* (Portland, OR, Sept. Sept. 2004). ACM Press, New York, 2004, 144–151.
2. Claypool, M., Claypool, K., and Damaa, F. The effects of frame rate and resolution on users playing first-person shooter games. In *Proceedings of ACM/SPIE Multimedia Computing and Networking Conference* (San Jose, CA, Jan.). International Society for Optical Engineering, 2006.
3. Claypool, M. The effect of latency on user performance in real-time strategy games. *Elsevier Computer Networks 49,* 1 (Sept. 2005), 52–70.
4. Dick, M., Wellnitz, O., and Wolf, L. Analysis of factors affecting players' performance and perception in multiplayer games. In *Proceedings of the Fourth ACM Network and System Support for Games (NetGames) Workshop* (Hawthorne, NY, Oct. 10–11). ACM Press, New York, 2005.
5. Fritsch, T., Ritter, H., and Schiller, J. The effect of latency and network limitations on MMORPGs: A field study of Everquest 2. In *Proceedings of the Fourth ACM Network and System Support for Games (NetGames) Workshop* (Hawthorne, NY, Oct. 10–11). ACM Press, New York, 2005.
6. Henderson, T. and Bhatti, S. Networked games: A QoS-sensitive application for QoS-insensitive users? In *Proceedings of ACM SIGCOMM Workshop on Revisiting IP QoS* (Karlsruhe, Germany, Aug.). ACM Press, New York, 2003.
7. Jaiswal, S., Iannaccone, G., Diot, C., Kurose, J., and Towsley, D. Inferring TCP connection characteristics through passive measurements. In *Proceedings of IEEE Infocom* (Hong Kong, China, Apr.). IEEE Computer Society, Piscataway, NJ, 2004.
8. Jehaes, T., Vleeschauwer, D., Coppens, T., Doorselaer, B., Deckers, E., Naudts, W., Spruyt, K., and Smets, R. Access network delay in networked games. In *Proceedings of the ACM Network and System Support for Games (NetGames) Workshop* (Redwood City, CA, May 22–23). ACM Press, New York, 2003.
9. Nichols, J. and Claypool, M. The effects of latency on online Madden NFL Football. In *Proceedings of the 14th ACM International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)* (Kinsale, County Cork, Ireland, June). ACM Press, New York, 2004.
10. Pantel L. and Wolf, L. On the impact of delay on real-time multiplayer games. In *Proceedings of the ACM International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)* (Miami, May). ACM Press, New York, 2002.

**Mark Claypool** (claypool@cs.wpi.edu) is an associate professor of computer science and director of interactive media and game development at Worcester Polytechnic Institute, Worcester, MA.
**Kajal Claypool** (kajal.claypool@oracle.com) is a principal member of the technical staff at Oracle Corporation, Nashua, NH.