Measuring The Latency of Cloud Gaming Systems

雷欽隆

台大電機系

E-mail: lei@cc.ee.ntu.edu.tw

http://crypto.ee.ntu.edu.tw

Outline

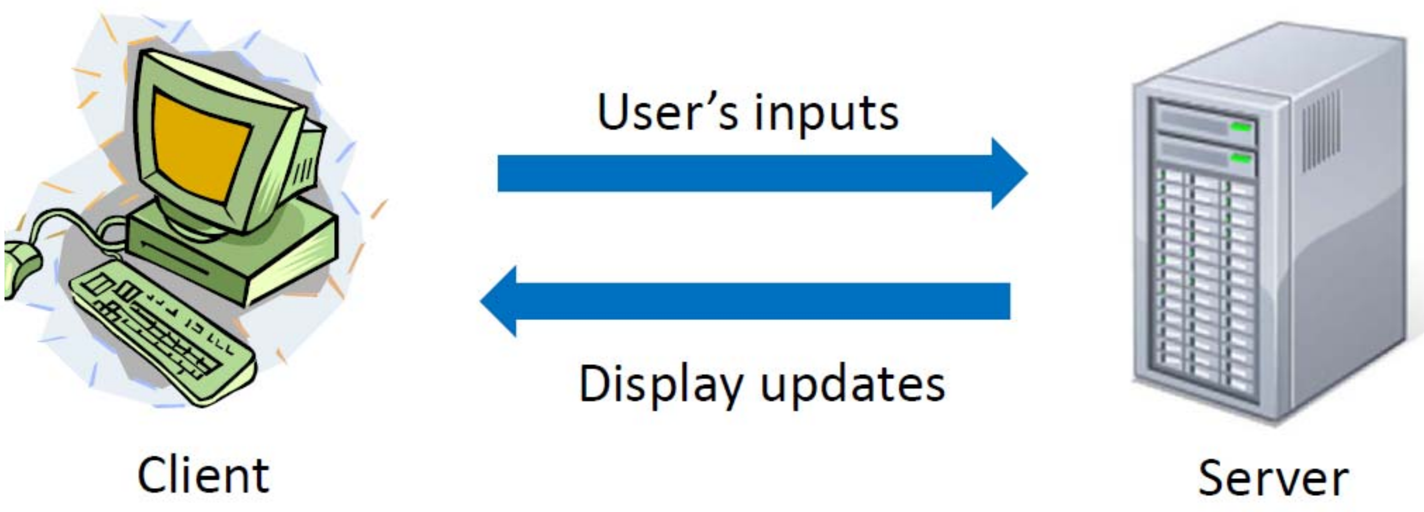
- Introduction
- Methodology: latency measurement
- Experiment setup
- Measurement results
- Conclusion

Gaming

- Computer games
  - Game software becomes more complex
    - The overhead of setting up a game is significant
  - Games have software and hardware compatibility with computers (players may be faced with a dilemma of whether to upgrade their computers or forgo the opportunity to try a new game)

Introduction

- Thin Client
  - People access and work in the remote servers via thin clients



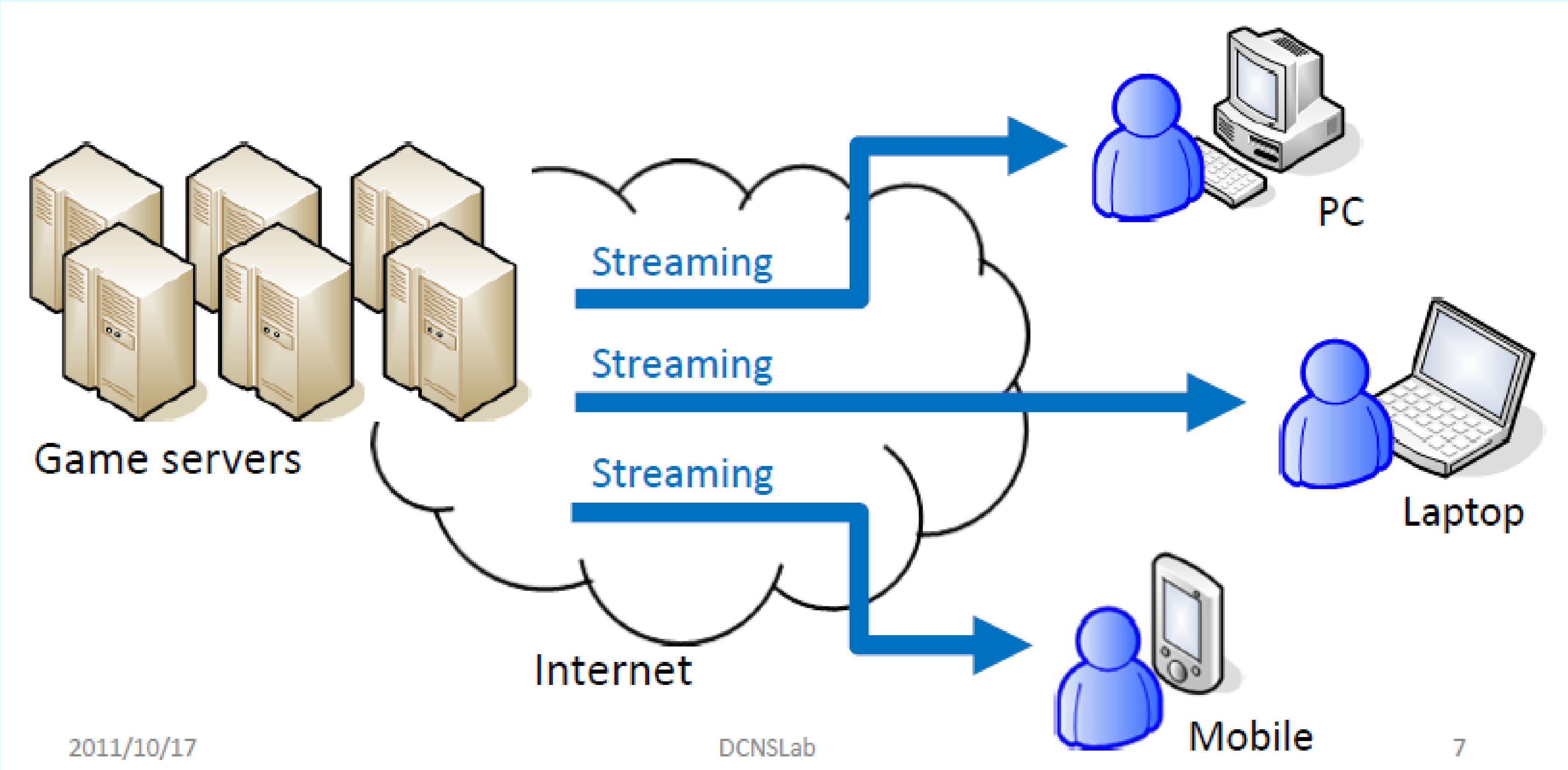
- "Understanding The Performance of Thin−Client Gaming," Yu−Chun Chang, Po−Han Tseng, Kuan−Ta Chen, and Chin−Laung Lei, IEEE CQR 2011.

What is Cloud Gaming

- Also called gaming on demand
- A type of online gaming that allows direct and on-demand streaming of games onto a computer through the use of a thin client
- The actual game is stored on the operator's or game company's server.
- The controls and button presses from the user are transmitted directly to the server, where they are recorded, and the server then sends back the game's response to the input controls.
- A low-level internet connection will also work (with only a "DSL connection of 1.5 mbps" needed for a standard-definition television.)

Cloud Gaming

- Real−time game playing via thin clients



Advantages of Cloud Gaming

- No set up overheads or compatibility issues
- People can play games anywhere, anytime
- Does not use local disk storage
- Thin client
- Platform independent
- Always the latest version
- Cloud-based video games will put pirates out of business

Cloud Gaming Services

GAIKAI

onlive

G-cluster Global

T5-Labs

StreamMyGame

otoy

Web Sites

- http://www.onlive.com/
- http://www.streammygame.com/
- http://www.gaikai.com/
- http://www.gcluster.com/
- http://www.otoy.com/
- http://www.t5labs.com/

G-cluster

- Cloud gaming as a gaming-on-demand technology was originally implemented in Finland by a company called G-cluster, founded in 2000 and a wholly owned subsidiary of SoftBank since 2004.
- The first commercial roll out of the service was for Cyprus Telecom Authority in 2005 on their IPTV network
- In March 2008 G-cluster announced HD support in their solution in with Amino IPTV STB.
- In November 2010, after thorough beta testing, French operator SFR launched a commercial service based on G-cluster technology.
- G-cluster uses IPTV set-top-boxes and the target audience is gamers that already have gaming consoles, as well as the entire family, with a game selection consisting of casual games as well as high end titles.
- G-cluster business model was introduced in an article "Developing Cloud Business Models: A Case Study on Cloud Gaming" of the IEEE Software Magazine July/August 2011 issue.

OnLive

- OnLive, Gaikai, and Playcast (gaming on demand companies) began beta testing in different parts of the world, such as Gaikai in Europe, in order to figure out the final configurations for their servers. (2009)
- Investors: Warner Bros, AT&T, Ubisoft, Atari, and HTC.
- It was released in June 2010 and now offers more than 120 games as of September 2011.
- OnLive’s client is available on Microsoft Windows, Mac OS X, and as a TV set-top box.
- Since the games are saved on the servers, users can access their saved game for any game they've played on the server "from any computer, at any time".
- The company also has a MicroConsole, which allows users to access OnLive with their television, so the games can be played on a screen even while being streamed from a server.

StreamMyGame

- StreamMyGame (SMG) offers software solutions for remote game playing.
- The service was launched in October 2007 and currently streams more than 120 games on Windows to Windows/Linux-based clients as of September 2011.
- It supports game streaming in a variety of resolutions, from 320x240 to 1920x1080 (1080p), which require an Internet connection between 256 Kbps (320x240) and 30 Mbps (1080p).
- StreamMyGame offers a cloud gaming platform rather than a centralized service.

Gaikai

- Gaikai, founded in 2008, streams game demos across 12 countries in Europe and North America, offering consumers to play games for free in a **web browser**.
- Gaikai reduces the time to play demos by eliminating form filling, credit card details, downloads and installs, making games playable directly inside web browsers via existing plug-ins or on devices like digital TVs.
- Gaikai gained a big burst of publicity in May 2010 when CEO David Perry showed World of Warcraft (魔獸世界) running on an iPad, using the service.
- Gaikai's cloud platform is an interactive entertainment network for gamers, publishers, retailers, and affiliates. The service consists of the Gaikai Affiliate Network and partnerships with video game publishers, developers, retailers, and hardware manufacturers.

Gaikai

- Websites sign up for the Gaikai Affiliate Network to stream video games from Gaikai's publishers via advertising running on their websites, in exchange for a cut of the revenue.
- Games also stream from game destinations powered by the video game publishing, developer, retailer, and manufacturer partners. Video game players who try out these games are given options at the end of the demo to purchase the game from a retailer, direct-to-drive download, or as a continued streaming service.
- Gaikai's gaming service went live in early 2011 by streaming Dead Space 2, The Sims 3, Spore, and Mass Effect 2,27 as well as announcing partnerships with Walmart, Gamecenter, Capcom, The Escapist, and Eurogamer.

Type of Games

- With cloud gaming, code processing and video rendering are done server-side as opposed to client-side.
- This allows any end-user or end-gamer to play any game on any system including old DOS/Apple/Commodore/Atari games, old arcade games, new arcade games, old console games, new console games, Windows games, Apple OS games, and phone app games.
- All the games can be patched on-the-fly by game developers. Game developers can code their games on any language and on any system they want; assuming that a given gaming cloud can process/render their game, it can be played by the end-user/end-gamer.
- Additionally, all games can be played with correct CPU timing (problem encountered when playing an old DOS game on a current CPU/microprocessor) and with highest graphics/physics settings at all times.
- Cheating and hacking may also be circumvented when all code is located on a remote server location

Limitations of Cloud Gaming

- The two main limiting factors: **network quality and bandwidth.**
- Network quality affects **response time and latency**.
  - **Server placement** of a given gaming cloud: Those end-users on a gaming cloud closest to a given server of the cloud would experience the least amount of lag or latency issues.
  - An end-user's/end-gamer's **ISP** also affects network quality, response time, and latency.
  - Ping time and "jitter" should be minimized as

Limitations of Cloud Gaming

- Bandwidth merely affects **quality of video stream**.
  - It's analogous to watching a movie on a ten-inch black & white TV or 1080p HD movie theater screen.
  - How much data that can be transmitted over bandwidth directly affects **quality and screensize** of video.
- Another limiting factor may be ability of local computer or system to properly render a video stream.
  - Video compression codecs and technologies may be used by a gaming cloud to reduce amount of data required to transfer.
  - It takes sufficient amount of processing power to **decompress and/or decode** such a video stream.
  - If data **encryption** is involved, it may also take processing power by local machine to decrypt..

Design Alternatives

- Some of the services are only accessible via thin clients on a PC (either native or browser-based applications)
- Some can be accessed via a TV with a set-top box.
- Design alternatives:
  - the way the existing game software is modified and run on the server;
  - the way the game screen is encoded (on the server) and decoded (on the client);
  - the way game streaming data is delivered to the client; and
  - The way short-term network instability is handled to maintain the game's responsiveness and graphical quality.

Which Service is Better?

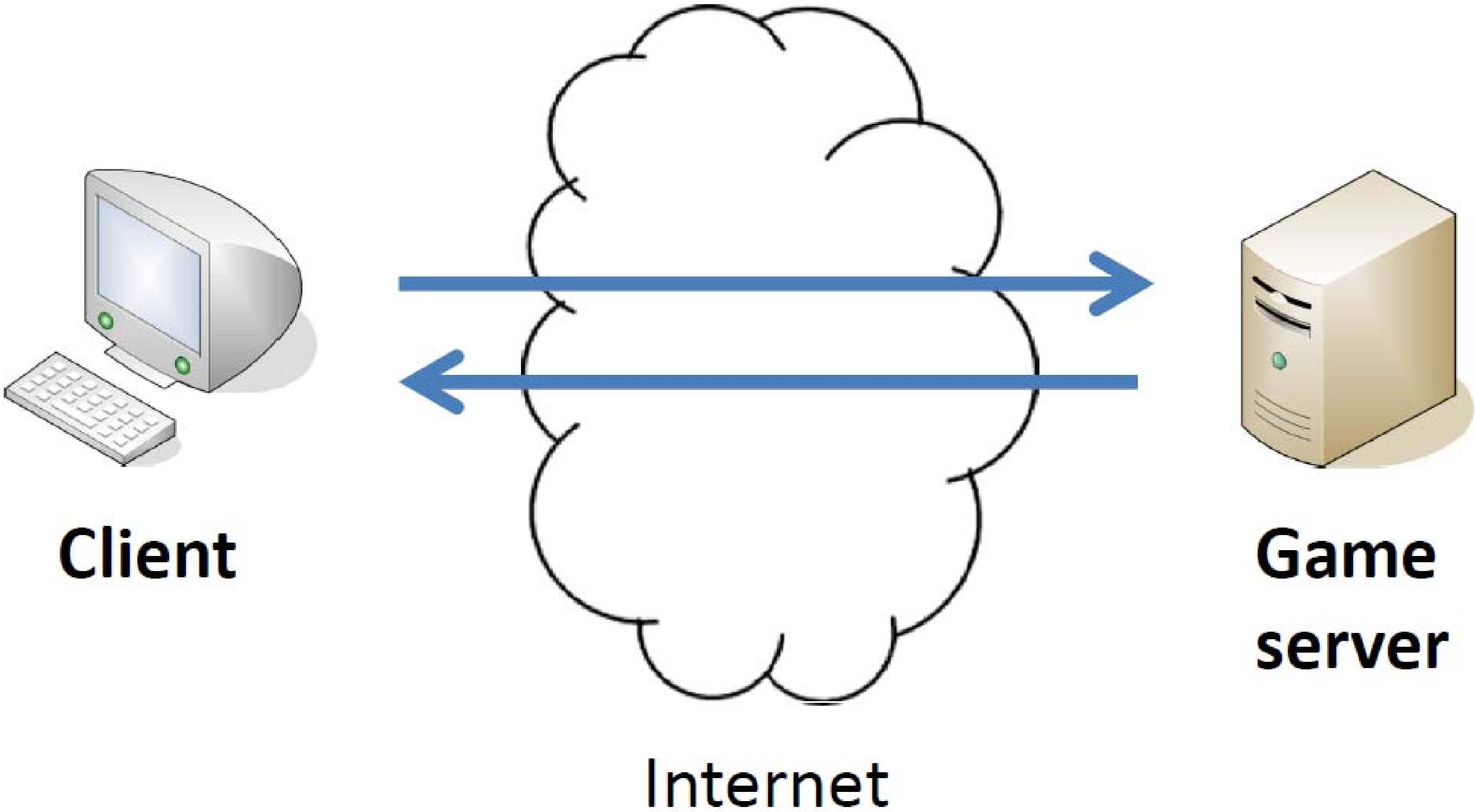
- Question
  - ***Which providers deliver the best service and which design elements constitute a good cloud gaming system?***
  - ***How good is the real-timeliness of current cloud gaming systems?***
    - **Response delay**

- Challenges
  - Most of cloud gaming systems are **proprietary and closed**
  - Game servers and the games running on them are managed by the service operators
- We propose a measurement methodology that can access the delay components of a cloud gaming system

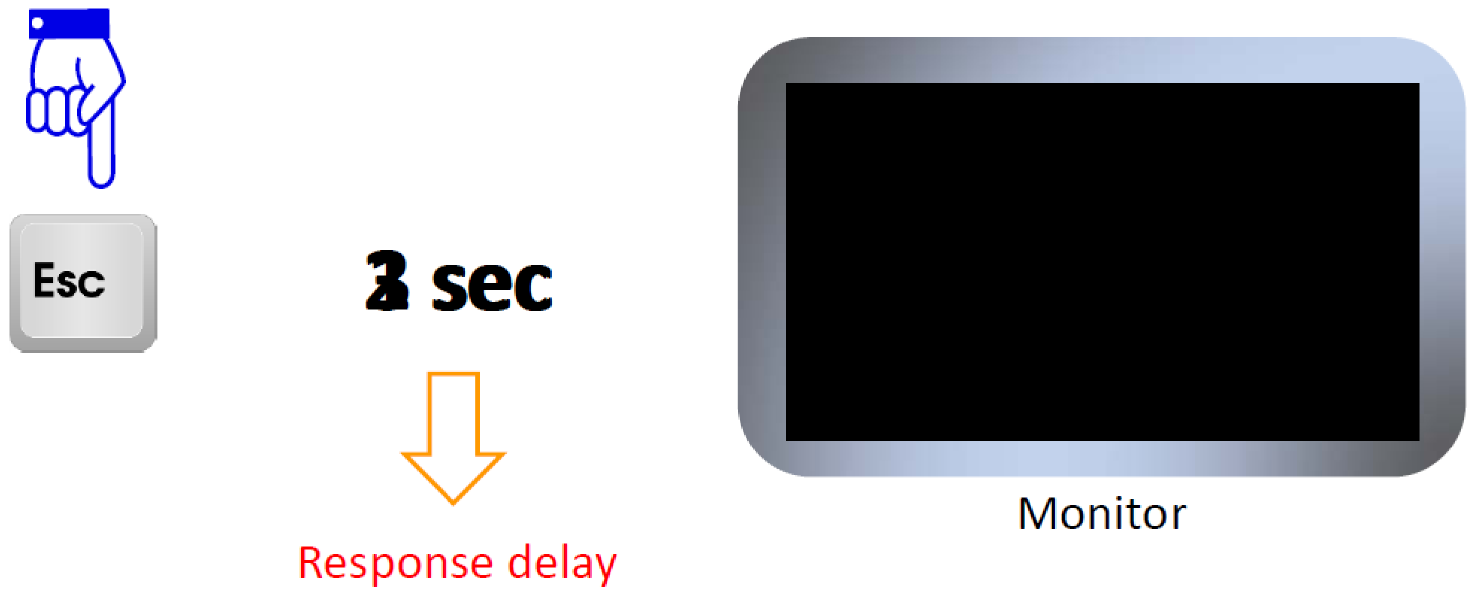
Related Work

* A. M. Lai and J. Nieh. On the performance of wide-area thin-client computing. *ACM Trans. Comput. Syst.*, 24:175–209, May 2006.
- Nieh and Laih proposed using slow-motion benchmarking to evaluate the performance of several thin client platforms on various tasks.
- Unfortunately, the technique cannot be applied to cloud gaming systems because remote applications would have to be modified so that they could run in slow motion.
- the performance metric used by the authors, i.e., the amount of data transferred, is not sufficient to accurately assess the temporal and spatial quality of cloud gaming services.

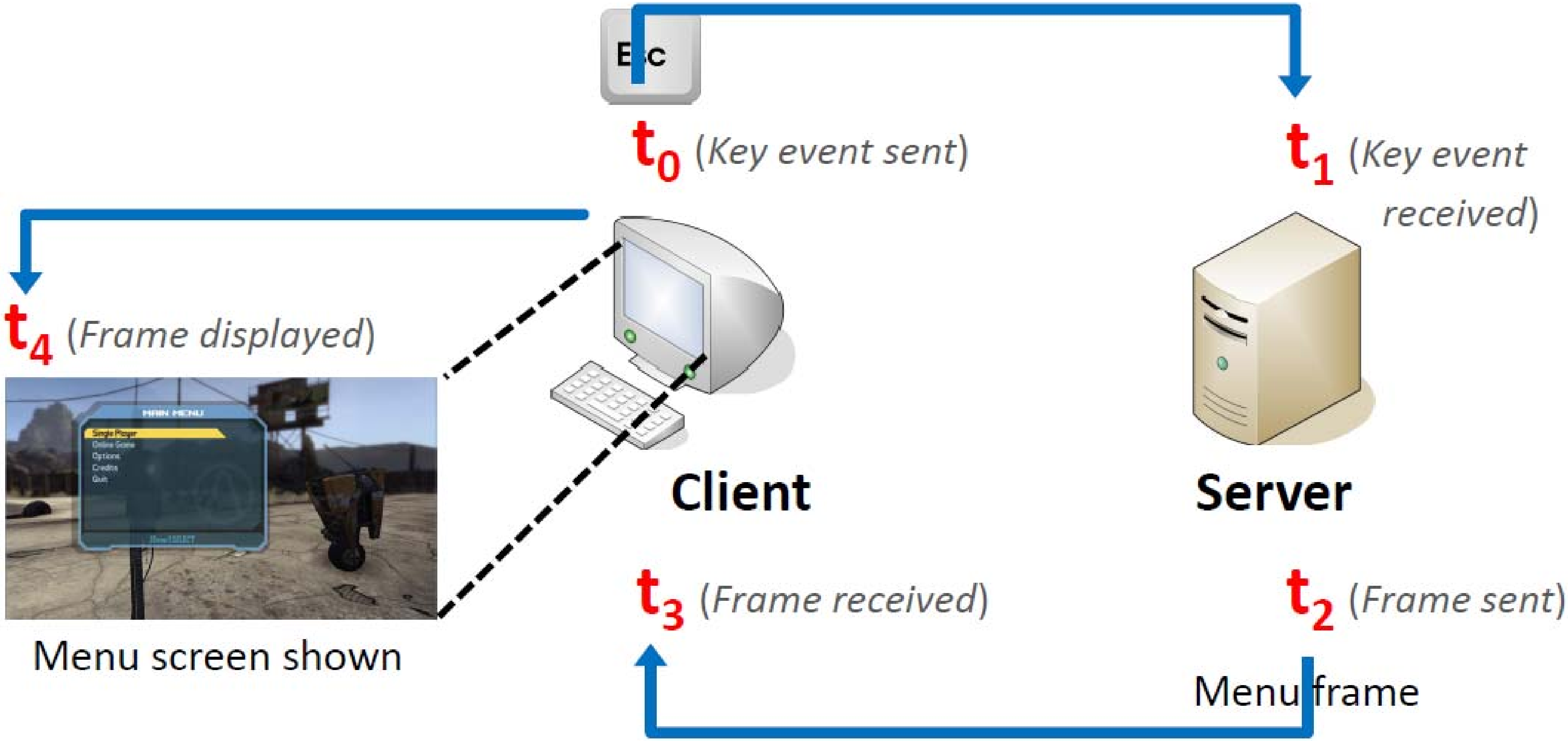
Cloud Gaming Systems



Response Delays of Cloud Gaming Systems



Response delay (RD) = Network delay (ND) + Processing delay (PD) + Playout delay (OD)



- Network delay (ND) : network RTT
- Processing delay (PD) : $t_2 - t_1 \rightarrow t_3 - t_0 - ND$
- Playout delay (OD) : $t_4 - t_3$

Determine the Response Delay

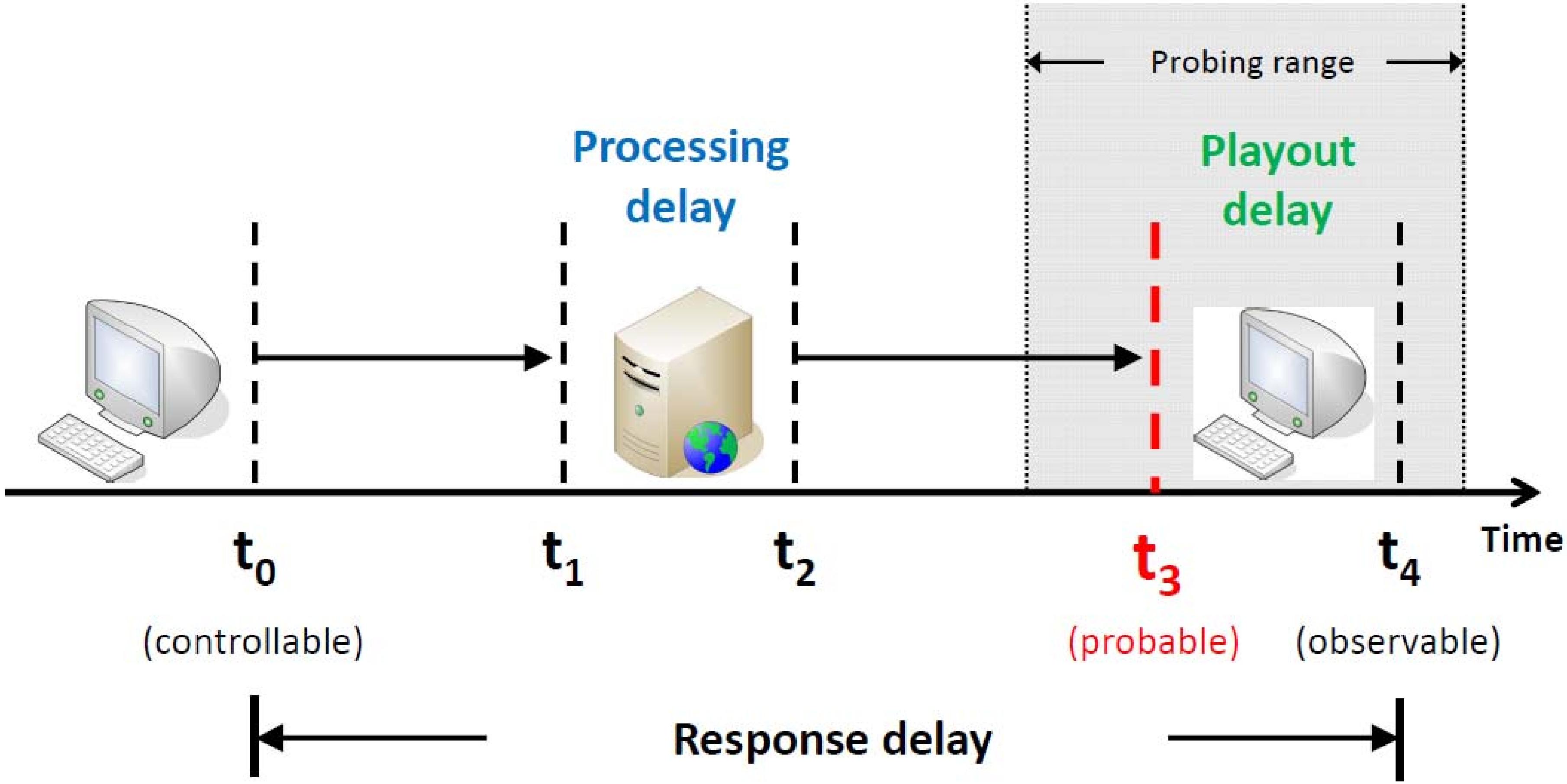
- The processing delay ($t_2 - t_1$) and the playout delay ($t_4 - t_3$) are not visible and cannot be measured directly.
- We utilize the hooking mechanism in Windows to inject our instrumentation code into the OnLive and SMG clients.
  - We use the detours library to intercept the **IDirect3dDevice9::EndScene() function**, which is called when a Direct3D application finishes drawing graphics on a hidden surface and is ready to present the surface on the screen.
  - We then use the following procedure to measure the response delays:

Measure the Response Delay

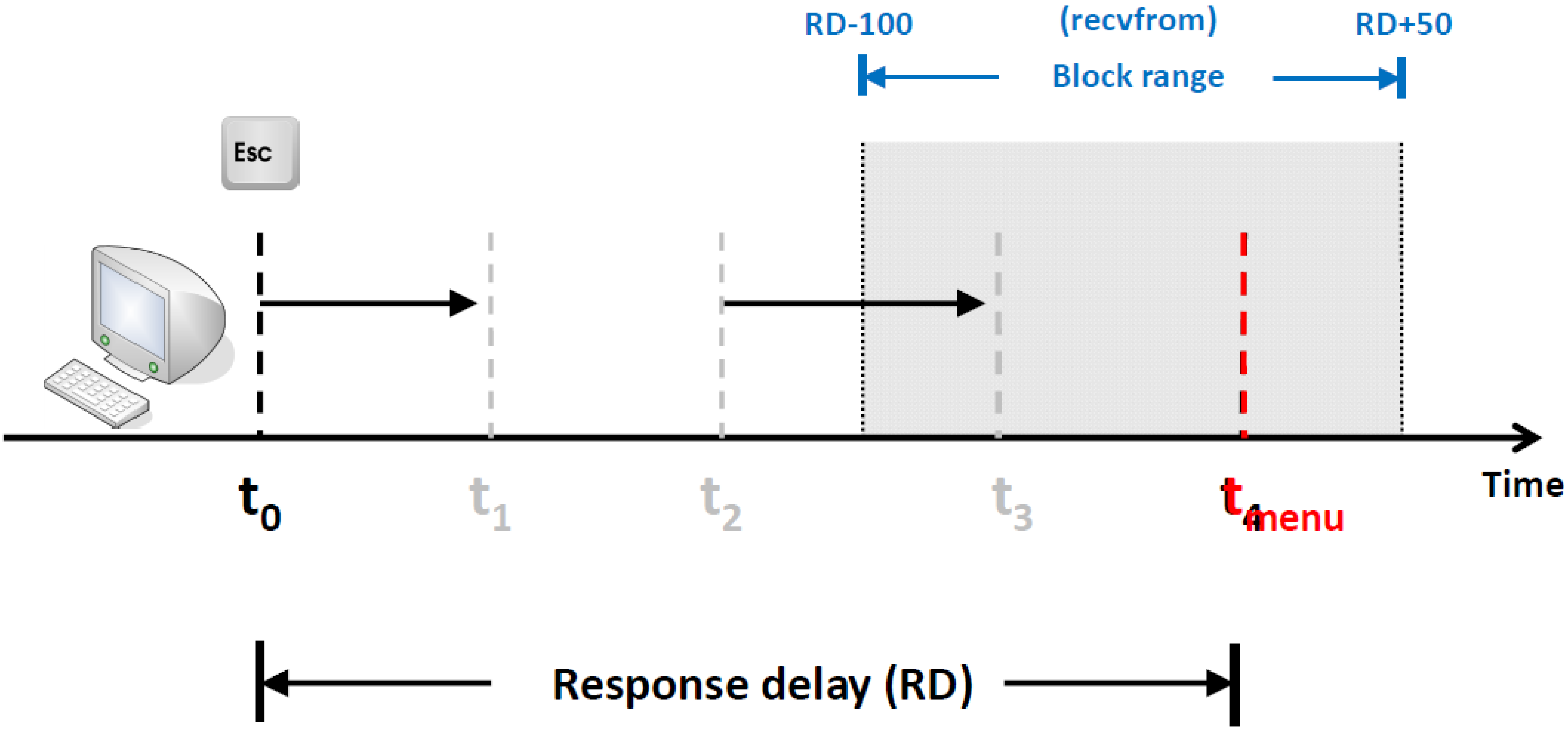
1. Send out an ESC key by calling the SendInput() function at time t_0.
2. Each time the game screen is updated, we examine the colors of a specific set of pixels to determine if the menu screen is displayed.
3. Wait until the menu screen appears (and note the time as t_4).

- Calculate the response delay by t_4 - t_0
- Each run of the procedure yields a sample of the response delay
- We can repeatedly execute the procedure to obtain a set of response delay samples.

Determine the occurrence of t_3



The Measurement Procedure



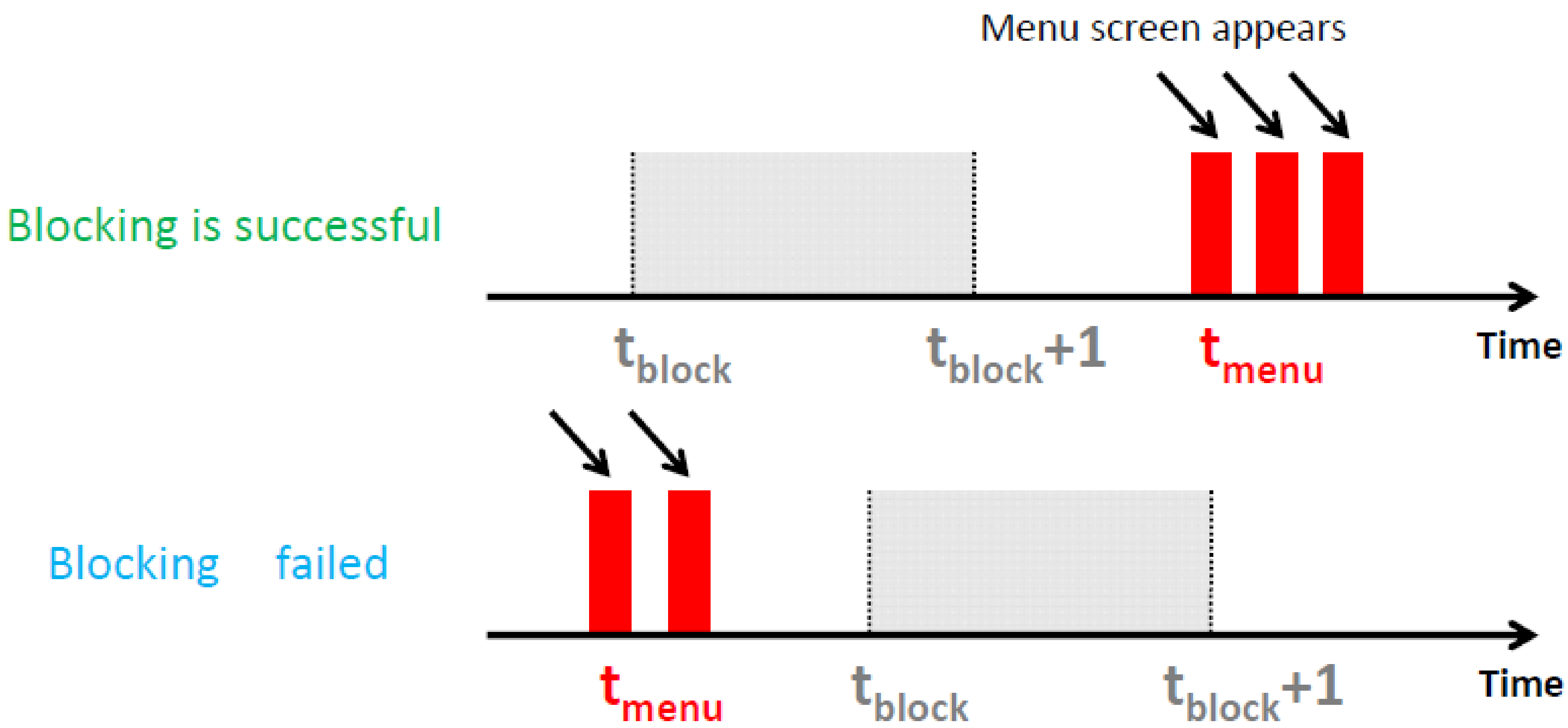
The Measurement Procedure

1. Simulate **ESC** key press event at time $\mathbf{t_0}$

2. Compute $\mathbf{t_{block}}$ as a random time [RD – 100, RD + 50] ms

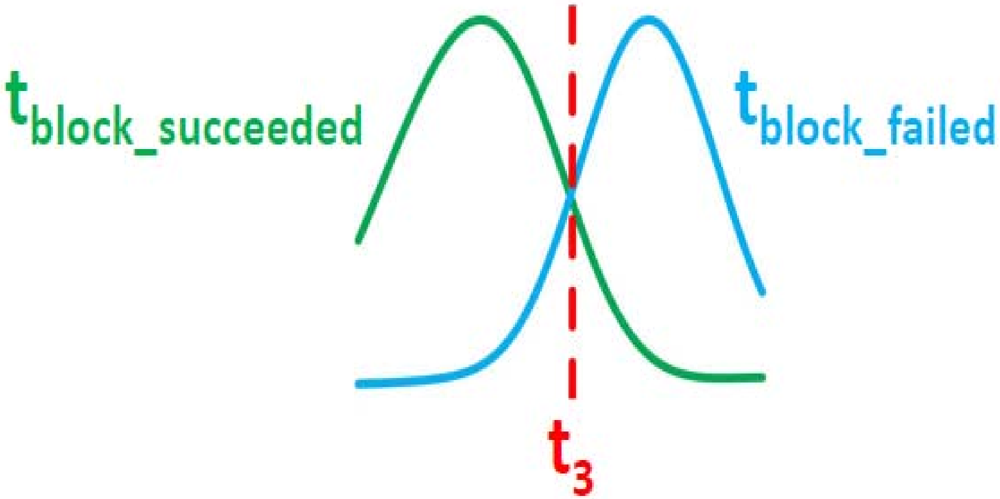
3. If the menu screen appears before $\mathbf{t_{block}}$, record the time as $\mathbf{t_{menu}}$. Otherwise, at $\mathbf{t_{block}}$, temporarily block ***recvfrom()*** calls for 1 second
   - recvfrom() is called when thin clients attempt to retrieve a UDP datagram from UDP/IP stack.

4. Wait until the menu screen appears (recorded as $\mathbf{t_{menu}}$)



- Blocking is successful : add t_{block} to the set $t_{block\_succeeded}$
- Blocking failed : add t_{menu} to the set t_{block_failed}
- t_3 would occur between the two sets
  - $t_{block\_succeeded} < t_3 < t_{block\_failed}$

- We then estimate t_3 as the point that yield the minimum sum of the two density functions formed by $t_{block\_succeeded}$ and t_{block_failed}

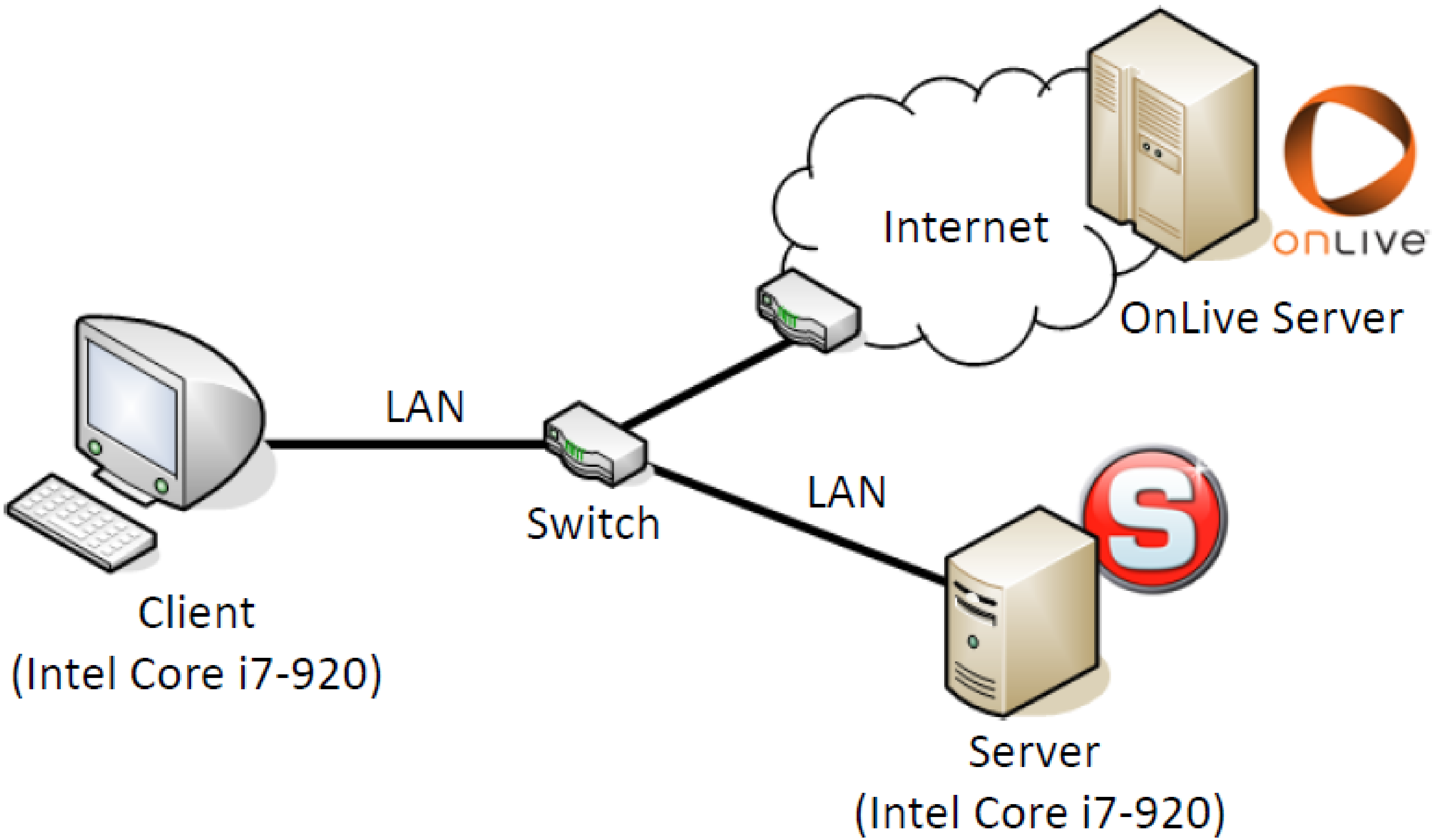


- Finally we compute
  - Processing delay (PD) = $t_3 - t_0$ – network delay (ND)
  - Playout delay (OD) = $t_4 - t_3$

Experiment Setup

- Platforms:
  - OnLive (**OnLive**)
  - StreamMyGame (**SMG**)
- Games:
  - Action : Lego Batman: The VideoGame (**Batman**)
  - RTS : Warhammer 40,000: Dawn of War (**DOW**)
  - FPS : F.E.A.R. 2: Project Origin (**FEAR**)

Experiment Setup

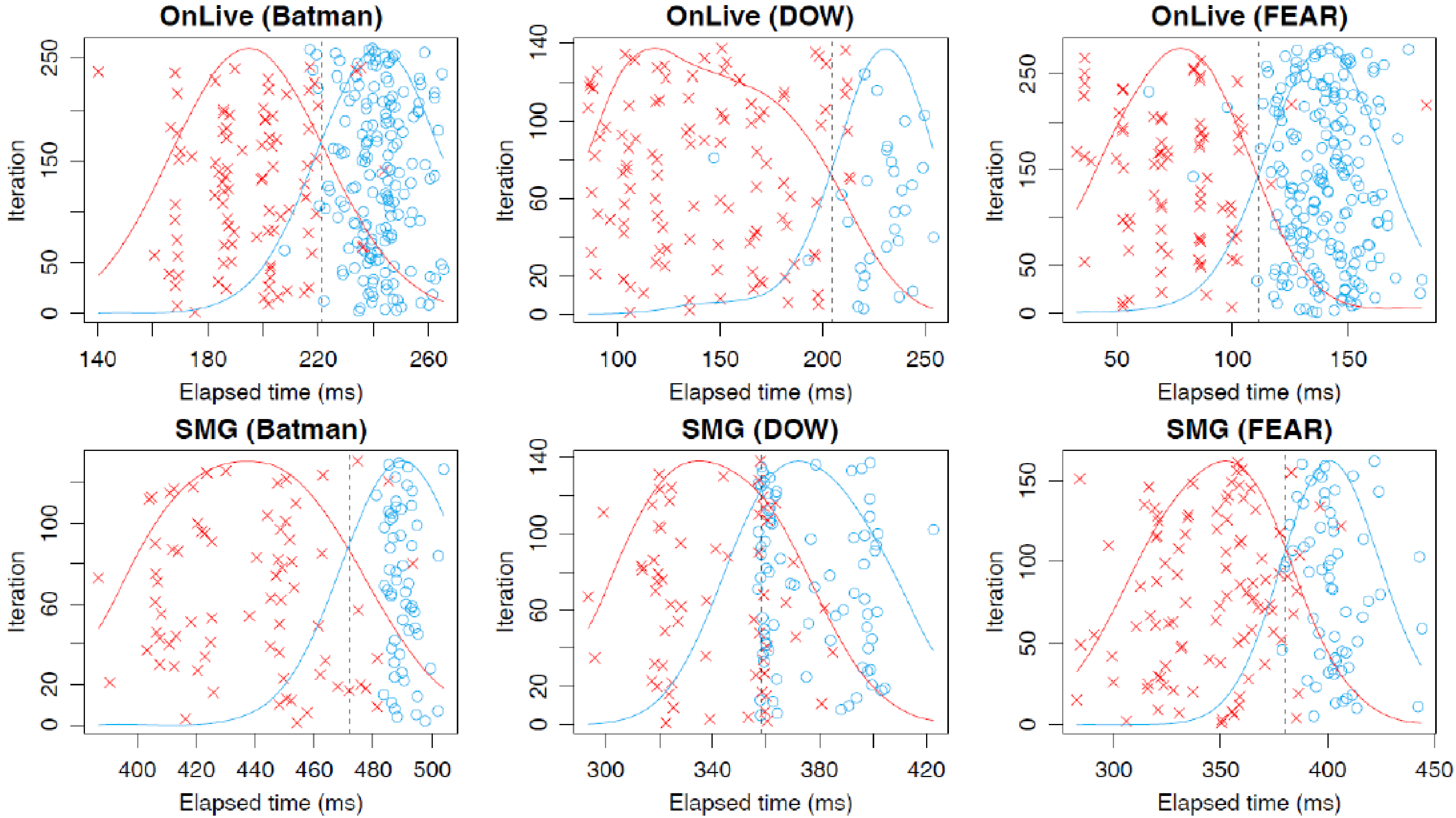


installed with the SMG server software

Experiment Setup

- The games are supported by both cloud gaming platforms
- They represent three game genres.
  - Lego Batman is an **action-adventure** game
  - FEAR is a typical **first-person shooter** (FPS) game
  - Warhammer is a **real-time strategy** (RTS) game.
- FPS games normally require a high rate of game screen updates
- The pace of adventure and RTS games is relatively slower
- These games are suitable for evaluating the general performance of both platforms because of their different workloads.

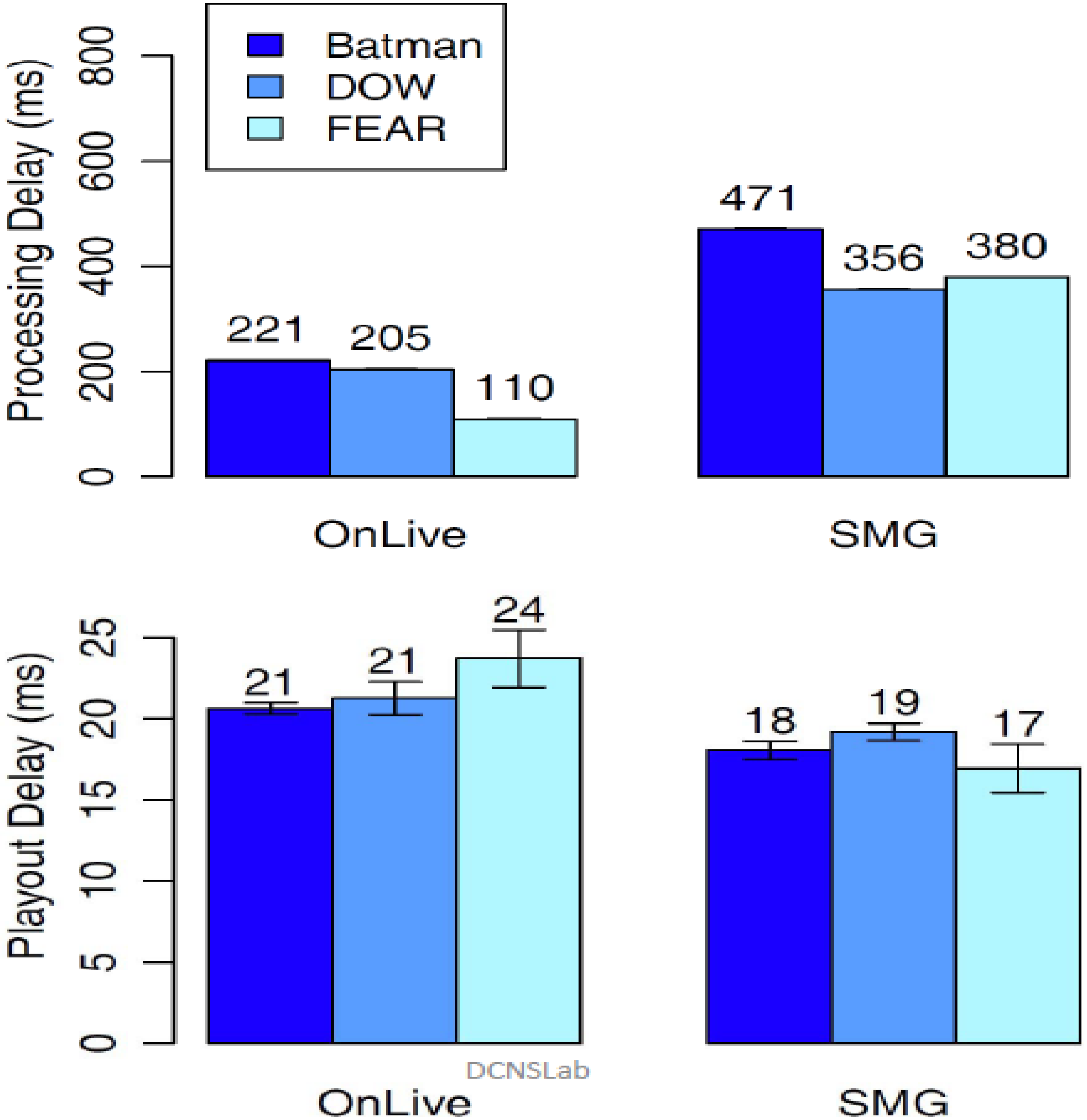
Experiment Results



X : $t_{block\_succeeded}$

O : t_{block_failed}[25]

Estimated Processing and Playout Delays



Conclusions

- We proposed a general methodology to measure the latency components of cloud gaming systems
- OnLive provides shorter latency for real−time gaming than SteamMyGame
- Our measurement approach can be generally applied to PC-based cloud gaming platforms, and it will further the understanding of such systems and lead to improvements
- Future works
  - Improve the proposed methodology
  - Apply the methodology to more platforms and derive
  - general guidelines for designing quality cloud gaming systems

Referwnces

- "**Measuring The Latency of Cloud Gaming Systems**," *Kuan–Ta Chen, Yu–Chun Chang, Po–Han Tseng, Chun–Ying Huang, and Chin–Laung Lei*, to appear in ACM Multimedia 2011.
- **"Understanding the Performance of Thin-Client Gaming,"** *Yu-Chun Chang, Kuan-Ta Chen, Po-Han Tseng, and Chin-Laung Lei*, May 10-12, 2011, Naples, Florida, USA.
- **"Online Game QoE Evaluation using Paired Comparisons,"** *Yu-Chun Chang, Kuan-Ta Chen, Chen-Chi Wu, Chien-Ju Ho, and Chin-Laung Lei*, IEEE CQR June 8–10, 2010, Vancouver, British Columbia, Canada.