Empirical Study of TCP and UDP protocols for gaming applications

—

Presented by: Maston Ho, Todd Xu

What is MMORPG?

- Massively Multiplayer Online Role Playing Game (MMORPG)
- Example of MMORPGs:
  - EverQuest (1999) - UDP
  - World of Warcraft (2004) - TCP
  - Blade and Soul (2012) - TCP
- MMORPGs require massive client-server communication
- Most modern MMORPGs use TCP as their transfer protocol

Introduction

- MMORPGs are one of the most profitable in game industry
  - World of Warcraft has about 800 million U.S. dollar profit in 2015
- Two main types of transport protocol: TCP and UDP
- We will be focusing on:
  - How game developers decide on which protocol they use?
  - Does the decision change overtime? Why and why not?
  - Reliability vs. Latency
  - Recommended solution

Characteristics of TCP / UDP

- Transmission Control Protocol (TCP)
  - Header size (empty packet) = 64 bytes
  - Three-way handshake
  - Acknowledge confirmation (ACK)
  - More reliable
  - Commonly used for web browsing and sending/receiving email
- User Datagram Protocol (UDP)
  - Header size (empty packet) = 52 bytes
  - More efficient
  - Commonly used for audio and video streaming

Why MMORPGs using TCP?

- UDP is more efficient with smaller packet size
- MMORPGs require massive and responsive transmission
- Possible reasons are …
  - Internet speed
  - Congestion control
  - Anti-cheat software

Internet Speed

- Internet speed improved and become affordable
- Packet drop rate remains low
- Able to support MMORPGs with TCP protocol without congestion
- Game developers prefer in reliability

Congestion Control

- Actual size of game packet is small
  - 46% bandwidth is occupied by TCP/IP header
- Nagle’s Algorithm
- But UDP could overwhelm
- Traffic is limited by application instead of network

Cheat Detection

Cheat detection relied on server side has evolved to client side

Example: Confirm explored fog of war

Consistency vs. TCP’s ACK

TCP Problem

- Retransmission timeout - ping increases significantly if packet dropped
- Big header - bandwidth or less latency?

Recommendations

- Disable Nagle's Algorithm
- Reduce server bandwidth on previous project JakeMUD (TCP-based MUD)
  - Movement detection
  - Send out a resume sequence number to client
  - Multiple streaming channels to process messages - Move, Attack, Talk
  - Move, Attack switch to UDP , Talk stick with TCP
  - Switch back to TCP with resume number from client

Conclusion

- All about trade-off
  - Infrastructure requirement vs. TCP’s reliability
  - Bandwidth vs. Speed
  - UDP unreliability vs. TCP retransmission

- Trend: TCP’s reliability, Speed, TCP retransmission?

Recent process

- Capture and analysis over a million packet transmissions using WireShark
  - We have tested: Blade and Soul, Fantasy Westward Journey, EverQuest
- Attempting to implement multi-streaming on JakeMUD
  - Hybrid application TCP vs. UDP protocol
- Study tradeoffs of TCP vs. UDP
  - How game developer decide?

References

Abdulazeez, S. A., Rhalibi, A. E., and Al-Jumeily, D. (2016). Simulation of Massively Multiplayer Online Games communication using OPNET custom application. 2016 IEEE Symposium on Computers and Communication (ISCC). doi:10.1109/iscc.2016.7543721

Chen, K., Huang, C., Huang, P., and Lei, C. (2006). An empirical evaluation of TCP performance in online games. Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology - ACE '06. doi:10.1145/1178823.1178830

Glazer, J. L., and Madhav, S. (2016). Multiplayer game programming: architecting networked games. New York, NY: Addison-Wesley.

Griwodz, C., and Halvorsen, P. (2006). The fun of using TCP for an MMORPG. Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video - NOSSDAV '06. doi:10.1145/1378191.1378193

Saldana, J. (2015). On the effectiveness of an optimization method for the traffic of TCP-based multiplayer online games. Multimedia Tools and Applications. doi:10.1007/s11042-015-3001-y

Ramakrishna, V., Robinson, M., Eustice, K., & Reiher, P. (n.d.). An active self-optimizing multiplayer gaming architecture. 2003 Autonomic Computing Workshop. doi:10.1109/acw.2003.1210202

Question?