Protocol Extraction for Blackbox-based Online Game Testing

Yeonjun Choi[1,2], Seongil Jin[2],

' ETRI, [2]Chungnam National University
Daej eon, Korea
june@etri.re.kr, sijin@cnu.ac.kr

Abstract. Test automation for online games can reduce testing time and cost because it doesn't need real massive users. Blackbox-based online game testing technology makes it easier to accomplish the goal. It should analyze the game protocols in the packets between game client and server. Change of protocols and scenarios on development and testing phase make it difficult to achieve the automation. In this paper, we introduce the extraction of game protocol to improve automation level of online game testing.

Keywords: online game, blackbox-based test, QA, MMORPG

1 Introduction

Online Personal Computer game market size is large and grows faster than ever. Especially MMOG(Massively Multiplayer Online Game) is attracted to games users with its rich variety of story and universe, social networking, beautiful graphic effects and realistic action of game characters.

MMOG game servers are usually distributed because of its massive number of concurrent user connections. Several issues in management of game servers arise like stability, tolerance, maximum performance and so on. In most of case, it can be evaluated by iterative testing with massive number of users game playing and it takes time and cost. Test automation can be a solution to reduce the cost[1].

We suggest the protocols of game packets for automated online game testing the rest of this paper.

1.1 Online game Testing

LoadRunner[2] provides packet replay-based load testing functionality. Prior to test it, you should provide exact packet information.

VENUS[3] is used to test online games extensible dummy virtual clients mimicking real gamer's action style, but you should also provide exact game packets and packet sequences.

Many changes during development and testing phase also make it difficult to automate game testing. MMORPG games are complicated application and

environment factors like massive users and internet connection quality affect the reliability and tolerance of online game.

1.2 Blackbox-based Testing

In this paper, we define two types of load testing as whitebox-based load testing and blackbox-based load testing.
Whitebox-based load testing means load testing technique with game contents embedding test code or recompilation in case of game protocol changes.

We can achieve automated testing with blackbox-based technique[4].

– Virtual clients with variable scenarios in groups. It enables us to write variable scenarios and apply them to the clients in groups.

– If any virtual client meet exceptional situation like meeting enemies that the scenario does not notify, the client automatically enter battle state because the client system already has battle action sequence.

– Virtual clients execute scenarios based on the analyzed protocols and real-time packet analysis.

2 Game Protocol Extraction

Game character play is a series of game protocols. Blackbox-based game test needs those protocols. Sometimes it's not easy for game QA testers to get game protocol details in a few reason like frequent protocol change. In this case, you need to extract game protocols from game play data.

The protocols of game, which are data packet transferred through network, are usually analyzed by human in following steps:

a) Separate header from the game packet
b) Extract information from game packet header, it indicates usually game data length and/or other data information
c) Extract protocol id from the data section for identifying the game packet.
d) Disassemble remaining data to extract the meaning of it.

Packet sequence analysis is processed after protocol data is analyzed. Simply with the protocol id, you can extract the sequence of game packet data and produce the basic game scenarios from the sequences.

Figure 1 show the automated steps of game packet-protocol analysis.

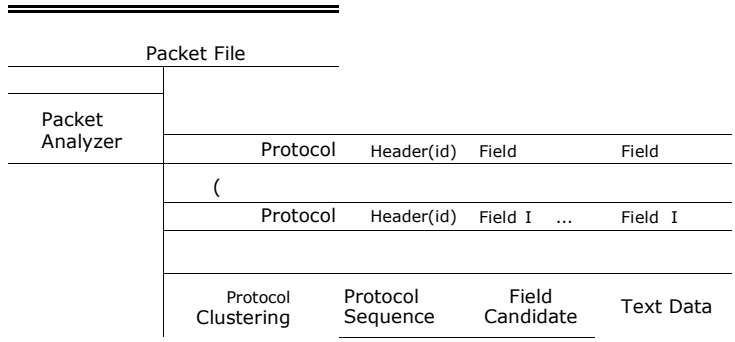

Fig. 1. Game packet analysis is processed through protocol extraction using clustering, sequence extraction, field analysis

2.1 Preprocessing

Before game protocol analysis, game packets should be preprocessed. After that game packets are clustered to extract protocols.

All input game packets should be preprocessed because packets are transformed to over-MTU(Maximum Transmission Unit) size packets and one-packed multiple packet for network transmission efficiency.

There are two kinds of transformed packets

- Packet is separated two or more network packets if the data size is bigger than systemically predefined MTU(Maximum Transfer Unit) size.

If data packet is too small, two or more data packets are packed in one packet for transmission efficiency.

In this step, we also considered game-specific data encryption of game data packet.

2.2 Clustering

Packet data is automatically classified using clustering.
After preprocessing step, data header is automatically extracted from data packet using data clustering.

We applied two kinds of methods to extract data header and protocol id.

1) Length-based Clustering

This simple idea is based on the similarity of the data packet length. If data size of two packets is same, they are probabilistically same protocol in practice.

But in many cases, same protocol data does not have same data length. The experimental accuracy was under 60%.

2) Hierarchical Clustering

It starts from simple idea.

a) The length of the data part is not flexible in many cases.

b) Protocol id appears almost same position

We applied hierarchical clustering to extract header parts and protocol id parts. The experimental accuracy was over 95% using our test game contents, ELMA.

Figure 2 shows clustered protocol sets.

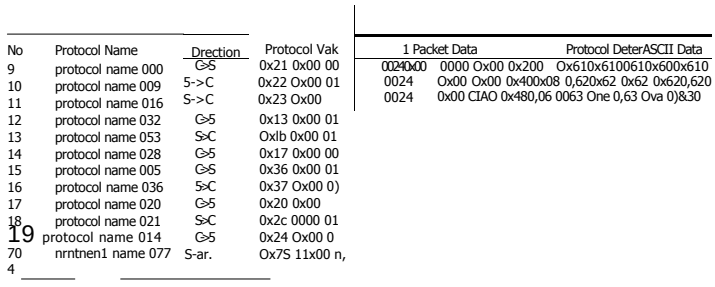

Fig. 2. Shows protocol clusters and each cluster means same protocol.

3 Conclusion

In this paper, we introduced Blackbox-based testing for massive online game testing. And we also introduced how automated data extraction is accomplished during game development and testing.

We developed Game packes and protocol analysis tools to capture and models the protocol for blackbox-based game testing. The result

References

1. Charles **P.** Schultz, Robert Bryant and Tim Langdell, Game Testing All in one, Thomson Course Technology PTR, 2005.
2. LoadRunner. [Online]. Available: http://www.hp.com
3. Bum Hyun Lim, Jin Ryong Kim and Kwang Hyun Shim, "A Load Testing Architecture for Network Virtual Environment," in Proceedings of 8th International Conference on Advanced Communication Technology, 20-22 Feb, 2006, pp. 848-852.
4. Blackbox and Scenario-Based Testing of Online Games Using Game Description Language, ETRI Journal, vol.33, no.3, June 2011, pp.470-473.