electronics

Article

Reliable User Datagram Protocol as a Solution to Latencies in Network Games

Jun-Ho Huh

Department of Software, Catholic University of Pusan, Geumjeong-gu, 57 Oryundae-ro, Busan 46252, Korea; 72networks@pukyong.ac.kr or 72networks@cup.ac.kr; Tel.: +82-51-510-0662

Received: 20 September 2018; Accepted: 31 October 2018; Published: 2 November 2018

Abstract: One of the major problems in network games has been that of latency (lagging) that game technology researchers are still tackling. Latency largely affects user satisfaction and it is often caused by insufficient hardware capacity or the internet speed that the user is employing. Even though online games are becoming more complex and the number of participants in these games is increasing continuously, users cannot properly deal with the requirements to play these games, as system upgrades and subscription changes to a higher speed internet service are costly. Instead of passing such a burden on to the users, the game companies should instead invest in providing an improved communication algorithm. Thus, the reliable user datagram protocol (RUDP)-based communication system design has been considered in this research instead of transmission control protocol (TCP) or user datagram protocol (UDP)-based systems. This could be a viable alternative system model. Many researchers agree that RUDP is a better protocol for the transport layer, but it seems that the large-scale testbeds that could support such an idea and carry out direct tests are very scarce, meaning that actual experimental implementation is difficult to achieve. This suggests that game designers or researchers need to rely on a small-scale testbed to collect performance data. Moreover, generating an analytic model from a small-scale testbed may not be viable due to the large number of elements involved in the implementation process, including latency. This study introduces an RUDP-based communication model created and tested by using the OPNET simulation tool. They are novel energy and cost-effective photonics technologies for access/metro networks.

Keywords: transport layer; RUDP; network games; OPNET simulation; leg; latency; transport layer architecture; computer architecture

1. Introduction

One of the major problems to be solved for network games is latency, which is usually referred to as 'lagging,' and affects the customer's satisfaction level greatly. Latency mainly results from an insufficient hardware specification or the internet speed of the user. As online games become larger in scale, a faster internet connection or better hardware specification is required, but it is quite difficult for users to upgrade their systems or change internet service providers. Therefore, it is better and more effective if game companies offer a better communication algorithm for their games. The reliable user datagram protocol (RUDP)-based communication system could be a good alternative in this respect, so a prospective model has been designed and implemented in this paper. The seriousness of the latency problem felt by game users is different depending on the games they play. That is, the players of real-time strategy (RTS) or turn-based games do not encounter this problem frequently because of the slower process of these games, but those who play games that require much faster reactions like twitchy or shooting games are affected by even a slight delay in their actions and therefore demand a faster communication system. The first thing to consider in regard to the latency problem is the communication distance. The longer the distance, the slower the communication speed. Gamers can check the communication response time by pinging a certain internet protocol (IP) address and the result obtained represents the communication environment as far as the speed is concerned. In a local area network (LAN) environment, this response can be obtained within a time range of between one and two milliseconds, whereas it will take a few to hundreds of milliseconds with an ordinary internet connection.

The second factor affecting game performance is the bandwidth used. Bandwidth indicates how much data can be transmitted in a fixed time, as well as the internet speed. For a LAN a speed range from 100 Mbps to 1 Gbps is usually available, while an average of 50 Mbps is guaranteed for common optical connections in the Republic of Korea (ROK). Figure 1 shows examples of the communication configuration for network games.

Figure 1. An example of the communication configuration for network games.

Finally, packet loss is another important factor determining game performance that should be dealt with. The volume of lost packets or undelivered packets is considered when evaluating the stability of the network communication. An increased number of packet loss incidents would mean that the network is unstable and that its quality is degrading.

There are several causes of packet loss but it mostly occurs due to overloads in the server, especially during gaming where many commands have to be processed simultaneously and instantly. When this happens, a user's PC screen may freeze or the communication with the remote game server could be interrupted. These phenomena are a little different from the slower movements in the massive multi-user online role-playing game (MMORPG) games caused by lower bandwidth.

The role of a user datagram protocol (UDP) in the network games is quite important for the reason that it shows a reaction speed close the physical network latency. In many cases, however, it is used restrictively due to its low data reliability. Nevertheless, if it is possible to improve reliability by approaching the problems in terms of supplementing its demerits, it will effectively replace part of the functional weak points of transmission control protocol (TCP). In this regard, many attempts have been made to implement reliable UDP (RUDP) for the well-known overseas games or network engines. It is expected that such a method will form a significant technological basis coupled with the activation of MMORPG.

2. Related Research

It has been a long time since the popularity of multi-player network-oriented games has languished while gamers' interest has widely shifted to the massive multi-player online games (MMPOG)/MMORPG. A proxy-based gaming architecture was proposed by Griwodz Carsten [1] who attempted to distinguish types of gaming traffic based on the urgency or relevance of each traffic type. This was to deal with the issues involved in scalability. Another similar architecture was proposed by [2] which was to reduce the traffic strain on the central game server. The average load on this server depends on the number of participating users has been compared, with [3] suggesting that peer-to-peer architecture effectively relieves stress (load) on the server, allowing a lower latency level than the fully centralized architecture of the server. Despite such merits, this architecture may not deal effectively with the inherent problems of state inconsistency resolution. Hence, the central arbiter architecture that assumes a hybrid role between the above two different architectures was proposed to take advantage of both strengths. During the past decade, a large number of studies were conducted to determine the negative aspects pertaining to jittering, delays, and packet losses in multi-player games or wireless games [4–10] based on the simulations. Some of them have focused on the implementation of the test bed environments, especially for wireless games [11–13].

One recent survey [14] clearly shows that network latency is one of the most critical elements relevant to user satisfaction in wireless games. An improved Quality of Service (QoS) means a better game performance based on descriptions [15] where it is been proven that the Universal Mobile Telecommunications System (UMTS) exhibits a better performance level in interactive real-time games than that of General Packet Radio Service (GPRS) as it, on some level, surmounts the problems of overprovisioning which often cause delays and jittering during the game process. It was suggested that statistical multiplexing and QoS together ensure an improved multiple game flow. More wireless gaming architectures have been introduced in recent years [16–18]. However, some of the present global leading wireless game providers are focusing on implementing their services on Enhanced General Packet Radio Service (EGPRS) that could be well-suited for the cross-layer design proposed in this study. Producing or launching wireless games with a reduced overhead is one of the major considerations which game providers should not take lightly as it would be costly to change the existing protocol stack for the intermediate nodes not connected to the game being played. Currently, either UDP or RUDP are often used for games [19], with the latter usually playing the role of continuously delivering the status of the current gaming information and also being suitable for lightweight game packet delivery. On the basis that TCP is a relatively slow protocol because of its complex congestion control algorithms and byte-oriented window scheme, this is considered a protocol that is not suited for wireless online games that accommodate multiple numbers of users simultaneously without causing delays.

In addition, TCP cannot even deal with moderate traffic congestion when delivering packets and this makes it quite clear that it is not at all suitable for MMPOG games. The connectionless protocol UDP, however, operates on the top of the IP layer and exchange data directly and recovers data when necessary, which seldom happens. As both are universal protocols widely adopted for most network devices and equipment, they are quite convenient for application developers. RUDP supplements UDP's error recovery function through a one-time acknowledgment scheme. Currently, both UDP and RUDP have become a major default protocol for wireless gaming infrastructures [20]. Other protocols including the game transport protocol (GTP) [21] have not been successful as their commercialization was always questioned. In the final analysis, it is clear that existing end-to-end QoS approaches for multimedia traffic do not work well with wireless games. As traffic generated during wireless games can be quite heavy depending on the number of participants and indicate an extreme value distribution [22,23], some new approaches that can effectively and efficiently deal with QoS-related issues should be devised to handle traffic [24].

2.1. Reliable User Datagram Protocol (RUDP) for Network Games

Communication protocols such as TCP and UDP are adopted in the transport layer by most network game companies due to their adaptability and universality in the design process [25,26]. However, although UDP is preferred more as its structure is relatively simple and offers a higher processing capability, its reliability is low as it does not check whether the packets have been delivered to the destination or not, and just keeps on transmitting the packets regardless. An additional problem is that sometimes packets will not be delivered sequentially. That is, ones that have been sent earlier than others could arrive at the destination faster. Figure 2 shows the RUDP for network games.

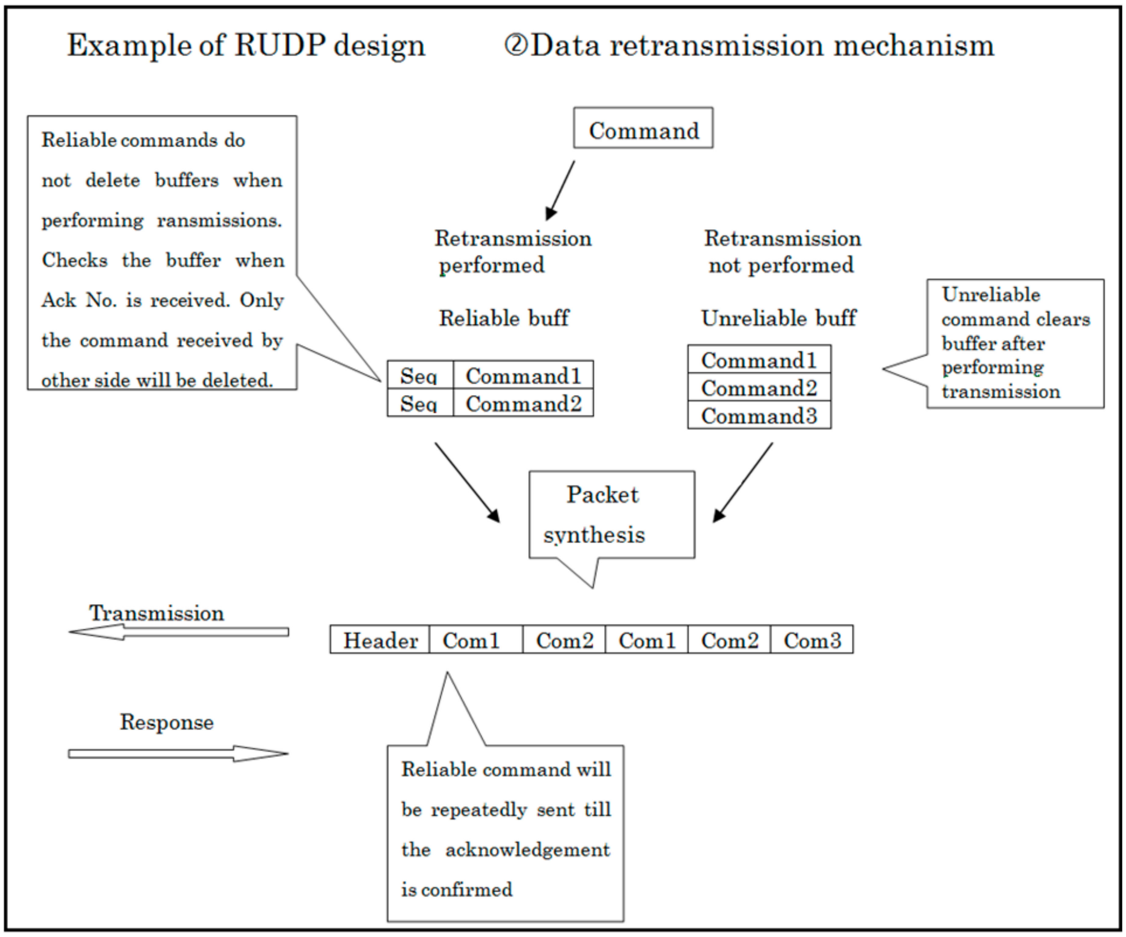
Example of RUDP design
②Data retransmission mechanism
Reliable commands do not delete buffers when performing ransmissions. Checks the buffer when Ack No. is received. Only the command received by other side will be deleted.
Command
Retransmission performed
Retransmission not performed
Reliable buff
Unreliable buff
Unreliable command clears buffer after performing transmission
Seq Command1
Seq Command2
Command1
Command2
Command3
Packet synthesis
Transmission
Header Com1 Com2 Com1 Com2 Com3
Response
Reliable command will be repeatedly sent till the acknowledgement is confirmed

Figure 2. The reliable user datagram protocol (RUDP) for network games.

On the other hand, TCP is relatively complex in structure and checks the packet delivery situation for re-transmission purposes. As it continuously checks delivery acknowledgement signals, its communication process becomes slower than that of UDP but offers higher reliability. In this regard, TCP is not suitable for network games but is adequate for routine internet operations such as mail transmission or web browsing.

Considering these factors, RUDP has become the first choice for network game companies since 2015. Smartphone-based game companies are also adopting RUDP as their standard communication protocol in the ROK. RUDP guarantees higher reliability by specifying the number of packets successfully delivered to the destination in the subsequent packet to be sent, and omitted packets will be re-transmitted accordingly [27,28].

Next, this paper will describe examples of performance by UDP and RUDP in an actual game play. The main techniques used in a fighting game called 'The Virtual Fighter 5 (VF5), Live Arena' are 'Perfect Synchronization' and 'Key-input synchronous communications', both of which assume that the players' screens are consistent with each other and that each key-input-data is transmitted to the opponent and synchronized by each frame. Synchronization can be achieved with simple key inputs if the game network is highly reliable or under the LAN environment, like the heavy fighting games where battles are engaged in every floor unit. Here data is transmitted through the UDP-based communication. It is important to check delays prior to initiating the games. Figure 3 shows a delay of approximately 40 ms until the data reaches the Two Frame (2F).

Figure 3. Example of Virtual Fighter 5 Using RUDP.

Based on this calculation, the Third Frame (3F) was set as a weight where synchronization will be achieved. In this structure, synchronization will be carried out by each frame. However, UDP does not always guarantee a complete key data delivery when a packet(s) is lost during communication. In such a case, the game will stop and wait for the next packet which will have the record of past data worth 10F. With this packet, the previous packet that has been processed will be dealt with. With the data worth 10F, one or two continuous packet losses can be recovered [25].

One-on-one fighting games like Virtual Fighter 5 will not often require a wide range of bandwidths but the multiple player games will have some problems [26]. That is, network games that invite multiple-fighters (e.g., 2 vs. 2, 3 vs. 3, or 4 vs. 4) will reduce the frequency of packet transmissions in order to enable much complex computing processes in large-scale games developed due to the evolution of smartphone games and interface upgrades, meaning that the adoption of RUDP should be considered.

2.2. Communication Types of Network Games

Today, TCP and UDP are widely used transmission protocols for both wired and wireless networks. Even though they both guarantee sequential data delivery and provide a congestion control function [25–27], UDP displays better packet delivery performance (real-time delivery). Based on reliable data protocol (RDP) [28–32], reliable UDP was proposed in the 1999 Internet Draft [3] as a transmission protocol that could offer reliability similar to TCP while preserving the low-delay characteristics of UDP [3]. Table 1 shows these communication types.

Table 1. Communication types.

User Datagram Protocol (UDP)	TCP	Reliable User Datagram Protocol (RUDP) (Ours)
Sometimes packet(s) may not be delivered.	Packet(s) will certainly be delivered.	Packet(s) will certainly be delivered.
Packet sequence may change.	Packet sequence is guaranteed.	Packet sequence depends on implementation.
Header size is small.28 bites including IP header.	Header size is large.40 bites including IP header.	Header size depends on implementation.
Speedy processing speed.	Slow processing speed.	Slow processing speed.
Likely to cause Network Address Translation (NAT) problem.	Unlikely to cause NAT problem.	Likely to cause NAT problem.

The main purpose of the simulations performed for the RUDP-based system in this study is to supplement the "unreliable" nature of UDP, which has been considered as its major problem, and thereby produce reliable data like TCP produces, by employing the better nature of UDP—not just imitating TCP's characteristics.

This study focuses on an RUDP-based communication system which has been proposed and supported by many researchers for its efficiency but has been difficult to validate as there has been no testbed large enough to conduct a direct experiment. So far the performance data of such a system can be collected from the small-scale testbeds only, but it is obvious that constructing an effective and adaptive analytical model based on these testbeds will not be possible due to the fact that some essential data collected from them would not reflect all the necessary elements required to generate a practical model.

The OPNET simulation tool was used to construct the RUDP-based communication system model proposed in this research, in which an advanced photonics technology had been adopted to establish a communication link with the metro networks.

Since UDP is basically a message-oriented and connectionless protocol, the network stream sent is not defined and relevant data will follow the pass through the router(s) operating under the less crowded network environment to process messages so that there is no process like the TCP's congestion control that controls the network flows [33–37]. If desired data is sent with RUDP, its transmission reliability can be guaranteed as much as TCP can offer as they will be retransmitted repetitively when transmission fails due to the poor network environment. However, here, a notable difference with TCP is that RUDP can solely use the network as it does not require any congestion control. The contribution of UDP in network games is great as it exhibits a reaction speed close to the physical network latency.

However, as the transmitted data is less reliable, UDP is often used in a limited way. UDP will be able to effectively achieve TCP-level reliability through establishing better reliability by compensating for the weakness of UDP more actively.

Such an implementation of RUDP has been attempted many times by famous overseas games and network engines and the author expects that it would become an important axis in technological development, along with the activation of MMORPG. RUDP is in fact being used by Korean game companies but they are not disclosing the details for security reasons. Although the following is not an unfamiliar subject, the sections below describe design and implementation phases of RUDP that focus on the utilization of UDP.

2.3. The Characteristics of User Datagram Protocol (UDP) in Network Games

User data protocol (UDP) has several notable characteristics compared to TCP:

- Data is transmitted or received unilaterally without giving any consideration to its connection.
- Transmission data is transmitted by the packet unit.
- There are no special processing processes for the transmitted data.

First, the fact that no thought has been given to the connection issue is its characteristic rather than its demerit. Nevertheless, considering that it is an element that burdens game programmers who need to consider or process connection problems, it could very well be a demerit.

From the system's point of view, each connection does not use a separate port and, as there is neither buffer or information regarding the connection, the system uses less resources. Therefore, since taking a resource security approach would not mean much, it will be correct to say that this is a characteristic rather than a merit.

Data are transmitted by the packet unit and their sizes are limited depending on the system settings such that this will not be a demerit for the games where mainly a small number of packet units are exchanged. Meanwhile, the fact that there are no processing processes for the transmission state or results is a notable characteristic.

- Data can be transformed.
- The order can be changed.
- Delivery may not be successful.
- Delivery can be made several times.

Although such characteristics are inherent to UDP, it would be more accurate to say that a network's intrinsic characteristics are as revealing as they are. Essentially, from the network level, it is also true that TCP will not be able to escape from such limitations but it does not need to bother with them as it compensates for these characteristics, whereas they will be burdensome to UDP users. Despite these demerits, the major strength of UDP is that it is faster than TCP as such error-correction processes are omitted. But, strictly speaking, the word 'faster' is somewhat incorrect.

Even so, it is not entirely incorrect also because, as for TCP, the data following the preceding data lost during its transmission must be on standby at the buffer and this leads to an additional delay to unidirectional network latency when the processed data actually arrives at the client side. For example, there are 4 data (A, B, C and D) and if A has been lost and the others have been transmitted, UDP will deliver B, C and D through Application Program Interface (API) first but TCP will not do so until A has been delivered through re-transmission(s). The contribution of this study is that the OPNET simulations have been performed to apply advantageous characteristic(s) of UDP to games while partially improving its reliability.

3. RUDP Implementation in the Transport Layer to Handle Latency

When RUDP is ideally implemented, it is possible to distinguish between the information essential and non-essential for playing games even if it has been lost. Also, the essential information can be distinguished into the cases where its order is guaranteed or unguaranteed.

Actually, since the amount and the frequency of these non-essential packets being lost during communications are low, they can be effective when they are used appropriately. The data which is to be transmitted repeatedly can be renewed/updated even if it is lost and the information which does not largely affect the visual component (e.g., the movement of an unnoticeable character at the edge of the screen, etc.) should be utilized more actively.

Unlike TCP, it is possible that the delay caused by momentary packet error in the network (latency) can be reduced a little.

For instance, supposing that the information (low-reliability + no order) of the characters A, B, and C have died in sequence has been transmitted by the server, it will not be so strange in the game that the characters B and C would die first followed by A when the information is used regardless of its credibility and order. In the case where TCP is used, the information of A will be retransmitted first and all the characters will die later. In other words, if the process is implemented with TCP only, the delay will not only occur for B and C but also for all the actions taking place after their death. Actually, UDP can be used widely in games as it has fewer constraints between the personal networks.

Meanwhile, RUDP is introducing the concept 'Reliable state' in addition to 3 data types. For example, when the state has been changed as A $\rightarrow$ B $\rightarrow$ C, any re-transmission request will not be requested even when only A has been received followed by C, skipping B. Therefore, RUDP is optimized for making the most of its characteristics of maintaining the latest information with reliability while maintaining the main concept of the game.

The game companies and their researchers are still dealing with one of the major issues in their gaming software, the latency or the lagging. A variety of factors (e.g., hardware capacities, internet connections, etc.) are involved in this issue and its effect is growing larger as the online games are becoming more complex and sophisticated allowing more participants to join them. Although the game companies encourage or recommend the users to upgrade their systems or change internet service providers to solve the problem, they still need to consider the situation where their customers would not be able to afford costly upgrades or move to an expensive internet service. This will definitely affect their sales policy so that they are consistently looking for a more efficient way of stabilizing the gaming speed without actually forcing the customers to make changes. One of the major problems involved with gaming speed has been the communication algorithm, and the communication protocol in particular.

3.1. A Design of RUDP Architecture

The core of this subject is compensating the negative characteristic of UDP, unreliability. However, the goal of this research is not just realizing the meritorious characteristics of TCP but also creating reliable data like TCP as well. The reliable characteristics of TCP compared to UDP can be summarized as follows. First, there are no data transformations. Second, there are no data losses. Third, data arrives in consecutive order.

Achieving all these points for implementation is the core theme in designing and implementing RUDP. Some possible methods for this purpose are: first, for the prevention of data transformation, it seems suitable to insert an error code in advance and check the error occurrence incidents. Then, ignore error packets or request a re-transmission.

It is possible to consider a method of recovering the errors by inserting a certain error-correction code, but this would also cause an unnecessary dissipation of bandwidth as the error rate in the UDP layer is actually very small. The negative characteristic related to data loss can be solved by persistently sending the undelivered packet until the opponent's acknowledgement packet arrives. It seems that this method has more merit in terms of processing difficulty or efficiency than letting the opponent transmit the packet-loss event packet, even if much dissipation of bandwidth is expected.

For the sequential processing of data, it is possible to process the data by checking the sequence of arriving packets first and tentatively storing the data subsequent to the unarrived packet during the transmission; then, sending out all the data to the game after all the preceding lost packets have arrived. Thus, RUDP is quite effective for latency control.

RUDP has a function of re-transmitting lost packets or receiving an acknowledgement packet. It is faster than TCP and utilizes UDP that guarantees packet deliveries with its convenient re-transmission structure. Nevertheless, RUDP is not a standard protocol provided like UDP or TCP. It has to be directly implemented in accordance with the protocol's specification or procured (i.e., separately offered in Software Development Kit (SDK). If it is possible to use the UDP implemented with the re-transmission function, the packets to be transmitted can be handled in the same way TCP does. That is, it becomes a protocol which can be used for any kind of game.

RUDP can be a good alternative for UDP when the game requires faster response times and higher reliability as it guarantees re-transmission of lost packets and 'guaranteed-order packet delivery'. Such a kind of selective-repeat protocol performs reliable communications by confirming the successful delivery of transmitted packets and also by implementing the timeout function. RUDP has been specifically designed to guarantee normal data flow. Below, Figure 4 shows the architecture of RUDP.

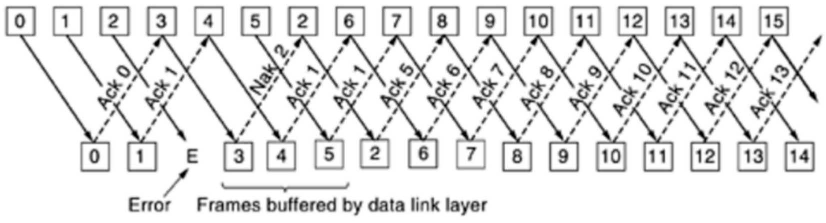

Figure 4. The architecture of RUDP.

Within Figure 4 is a description of the service RUDP is offering. The two-way transmission process is carried out in the application layer where data is transmitted through a reliable channel. Figure 4 briefly describes the RUDP service implementation process wherein a sender and a receiver maintain a pre-defined window size (i.e., usually maximum size) to avoid any errors in the communication process occurring from packet loss incidents. Even if the data has been sent over an unreliable channel, a higher rate of successful delivery can be expected. The architecture involving both sender and receiver is discussed in the other sections of this chapter.

In order to avoid a 'Deadlock' situation, the shared buffers were synchronized by employing the counting semaphores so that only a single thread can access the buffers each time. The respective window size (i.e., window sizes of sender and receiver) are traced consistently by observing the variables 'base' and 'next'. The variable 'next' will be increased by one when a single packet has been sent from the sender and the same increase will be observed for the variable 'base' when a single Ack (Acknowledgement) packet has been received by the sender.

This process is repeated continuously to check the number of packets contained in the buffer. The timeout is re-adjusted once the packet has been transmitted by using the timer. To test the performance of the implemented RUDP model, several packet loss and network delay scenarios have been simulated. The packets assigned with the value corresponding to the current time of the system plus the value of the network delay were aligned in a queue before being carried over to the socket. When the current time matches the value of a certain packet, that packet will be removed from the queue and transmitted after some delay. The packets also bear sequentially-assigned numbers so that the packets will be transmitted sequentially according the corresponding acknowledgement numbers.

3.2. Design of the Re-Transmission Function

It is a normal routine for the Receiver Node to send back an Ack packet to the Sender Node once it has received any data packet but if this procedure has not been completed, the Sender Node will repeatedly transmit the same packet following the communication protocol being adopted. Such a function is embedded in the system to guarantee the reliability of the data being transmitted.

$$Retransmission\ Timeout = \left(\frac{n-1}{2}+1\right)RTT \tag{1}$$

$$Max\ Retransmission\ Count = ceiling\left(\frac{Data\ Packet\ Aging}{Retransmission\ Timeout}\right) \tag{2}$$

The maximum cumulative Ack count and the average number of round trips are represented as n and Round-Trip Time (RTT) in the equation. The measurements are taken by the timestamp (extension) attached to the RUDP header. The number of re-transmissions will be smaller than that of the 'Go Back N' protocol or the 'One-Bit Sliding Window' protocol. One of the limitations of the Selective Repeat Protocol is the complexity of communications between the Sender and Receiver Nodes as each packet has to be acknowledged respectively. However, such a limitation is not a definitive obstacle considering the effectiveness and efficiency of RUDP over TCP or UDP. The Receiver Node returns an acknowledgment packet as a Positive Acknowledgment (P-Ack) when it has been received in a normal way along with its sequence number. This way, the RUDP will not have to use a Negative Acknowledgment (N-Ack) and avoid any complexity involved in the transmission process.

3.3. Node and Process Model for RUDP Using OPNET Simulation

Figures 5 and 6 represents the RUDP node model and the RUDP process model, respectively. The former was implemented by using app UDP module on Tpal. Layer 2 of this model was developed to use Medium-Access Control (MAC) and to carry out the TCP/IP transmission.

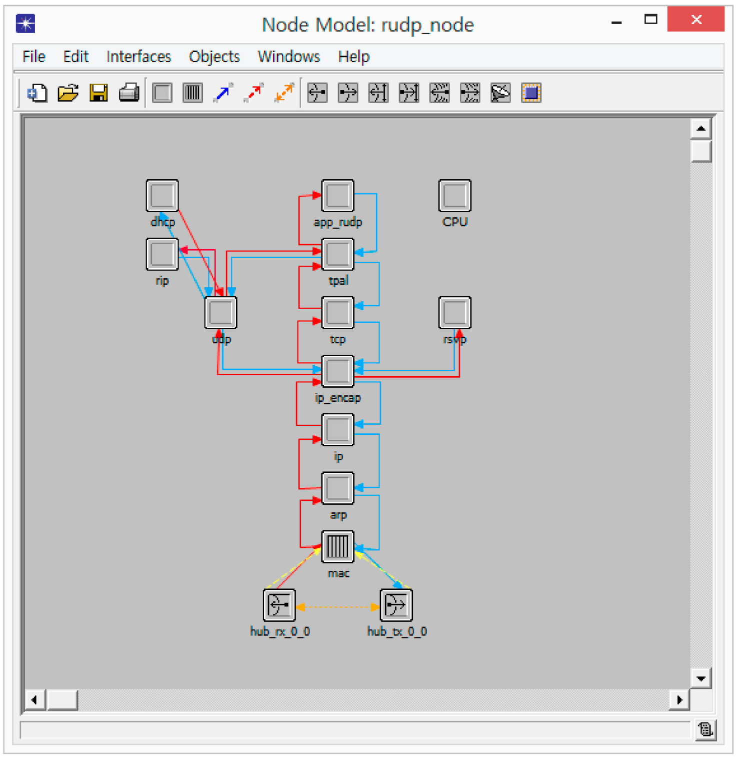

Figure 5. RUDP node model designed by using OPNET.

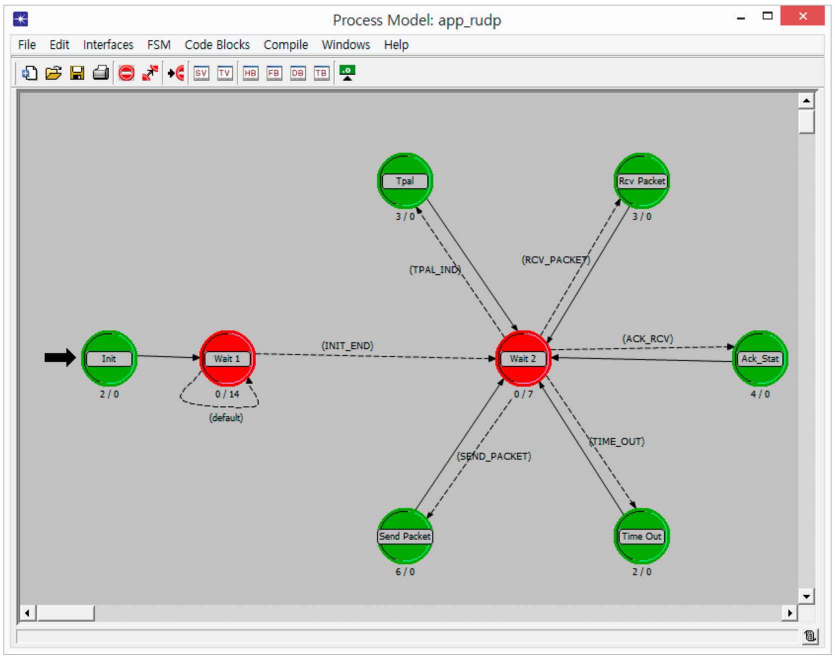

Figure 6. Implementation of RUDP process model designed by using OPNET.

The additional important attributes in the RUDP node model are the port numbers, the number of retransmissions, time-out periods and the transmission protocol. The RUDP process model was implemented to use all TCP, UDP and RUDP transmission protocols. For the RUDP model, a methodology that guarantees packet reliability by transmitting the packets in accordance with the 'Sequential (order) Numbers', and receiving the Ack packets after the message deliveries has been completed.

This is called the 'Selective Repeat Protocol'. If an Ack packet does not arrive within the designated time-out period, the same packet will be retransmitted until a fixed maximum number of retransmissions has been achieved. This model is expected to discard the retransmitting packet if it is unable to receive a relevant Ack packet in the end. The number of retransmissions and time-out periods can be pre-defined using the 'Property Window' such that the user can run the simulations by altering them each time.

4. Performance Evaluation

A test bed experiment has been conducted for the RUDP model by using the OPNET Modeler (14.5 PL8, OPNET Technologies/Riverbed, Bethesda, MD, USA, 2010) in Transport Layer and also the Microsoft Visual Studio (2013, Microsoft, Redmond, WA, USA, 2013) as a compiler. Interworking with Microsoft Visual Studio 2013 and OPNET Modeler 14.5 PL8, all the functions necessary for the node models for the simulations were coded and debugged with the C++ language.

4.1. Simulation Scenario Construction for the Network Games

Figure 7 shows a scenario concept for network games. The same concept applies to all game situations. As for the node arrangement, one client, one server, two switches and two routers have been deployed, and as for traffic, the author evaluated its performance by differentiating the transmission protocols (i.e., TCP, UDP, and RUDP), using the RUDP demand model. The simulation run time is one hour and the performance index has been determined with traffic data send/receive rates along with the end-to-end transmission delay time.

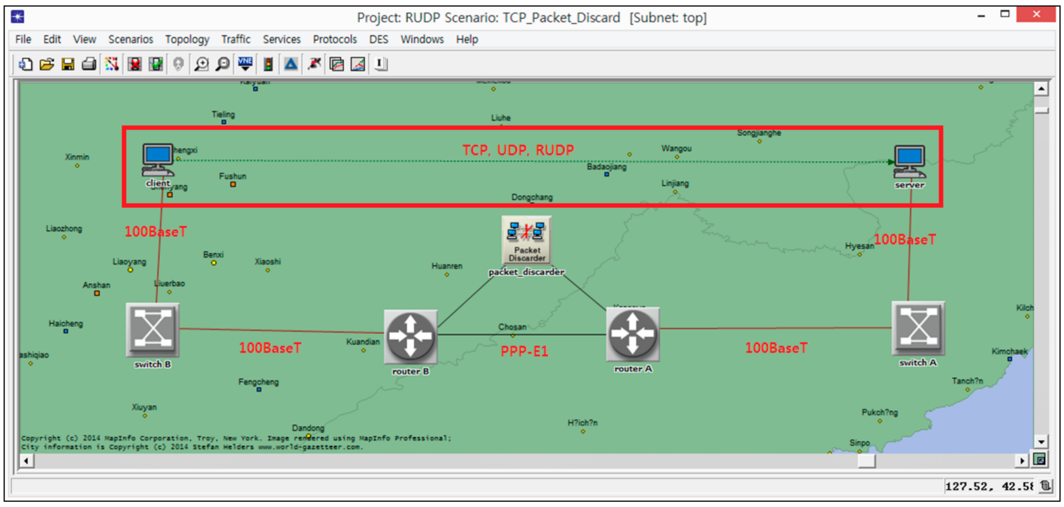

Figure 7. Scenario concept for network games.

4.1.1. Performance Evaluation of TCP Scenario

The following is the analysis result of the TCP performance in the scenario based on the traffic statistics: traffic speed (bits/s): the message transmission/receiving rate was the same as the traffic speed (300 bytes/s) which has been set initially and the entire performance analysis result is shown in Figure 8 which shows that all the messages have been received normally without any packet losses. This, and the absence of any background noises, shows that all the messages have been received correctly. As before, the end-to-end delays are calculated as the time required for a packet to reach the receiver node from the sender node.

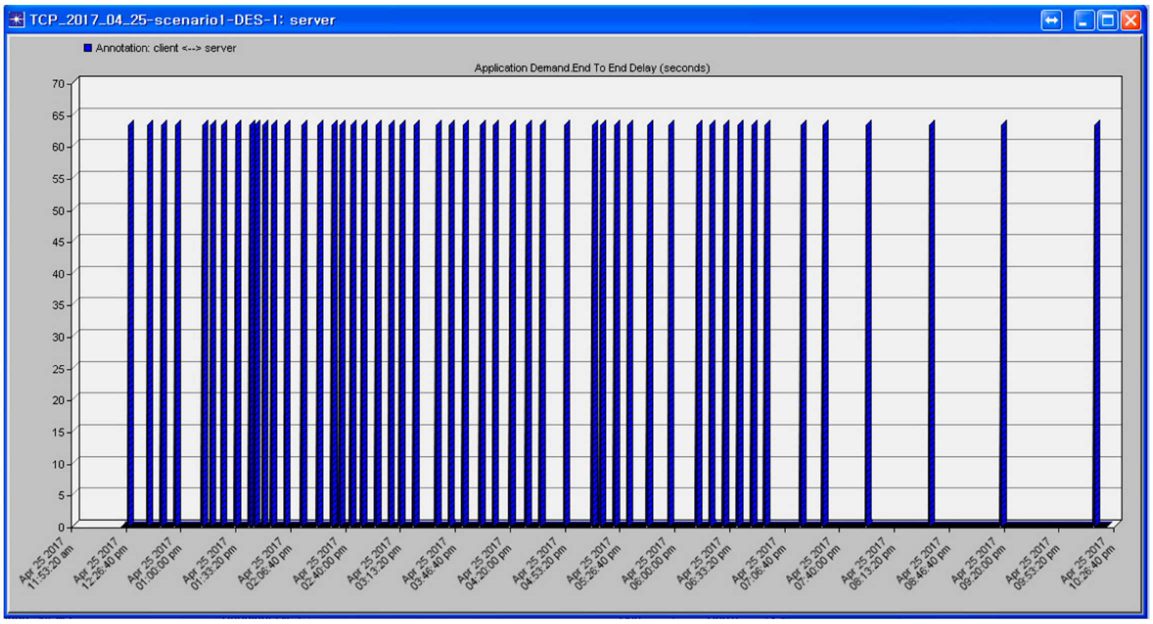

Figure 8. Results of TCP performance evaluation scenario.

4.1.2. Performance Evaluation of UDP Scenario

The analysis results of the UDP performance evaluation scenario using traffic statistics are as follows: traffic speed (bits/s): the message transmission/reception rate was the same as the traffic speed (300 bytes/s) which has been set initially and the entire performance analysis result is shown in Figure 9, which shows that all the messages have been received normally without any packet losses. This, and the absence of any background noises, also shows that all the messages have been received correctly. The average end-to-end delay during the packet transmission/reception was 0.0047 s.

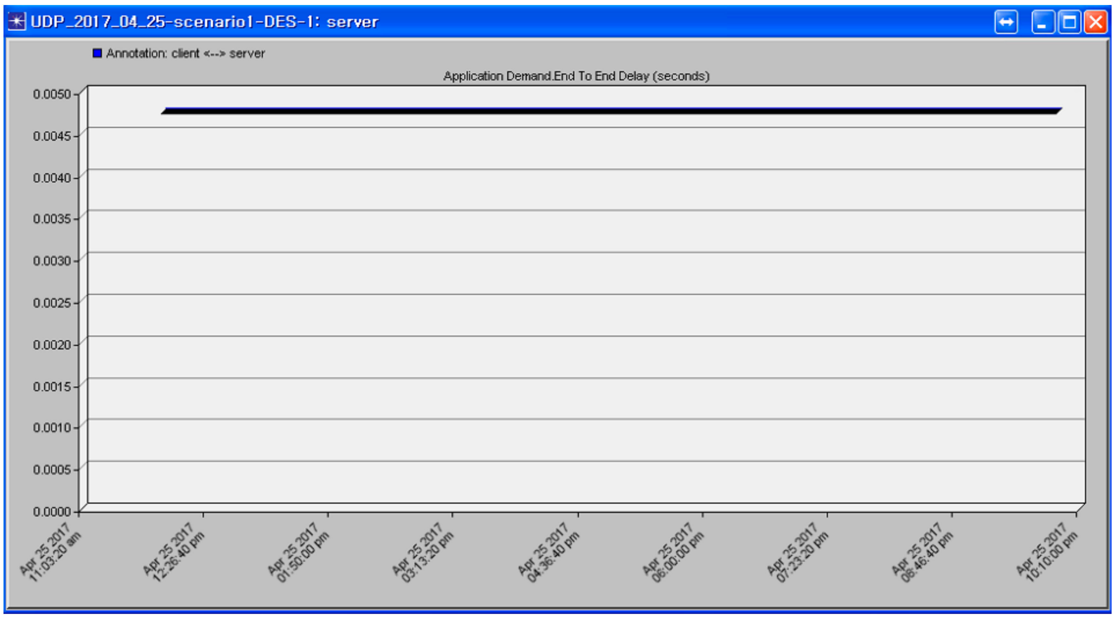

Figure 9. Results of UDP performance evaluation scenario.

4.1.3. Performance Evaluation of RUDP Scenario

In conditions for the performance evaluation of the RUDP were the same as the above two cases and the result is shown in Figure 10, which shows the same achievements but with a different average end-to-end delay of 0.020 s. (i.e., from the client node to the server node).

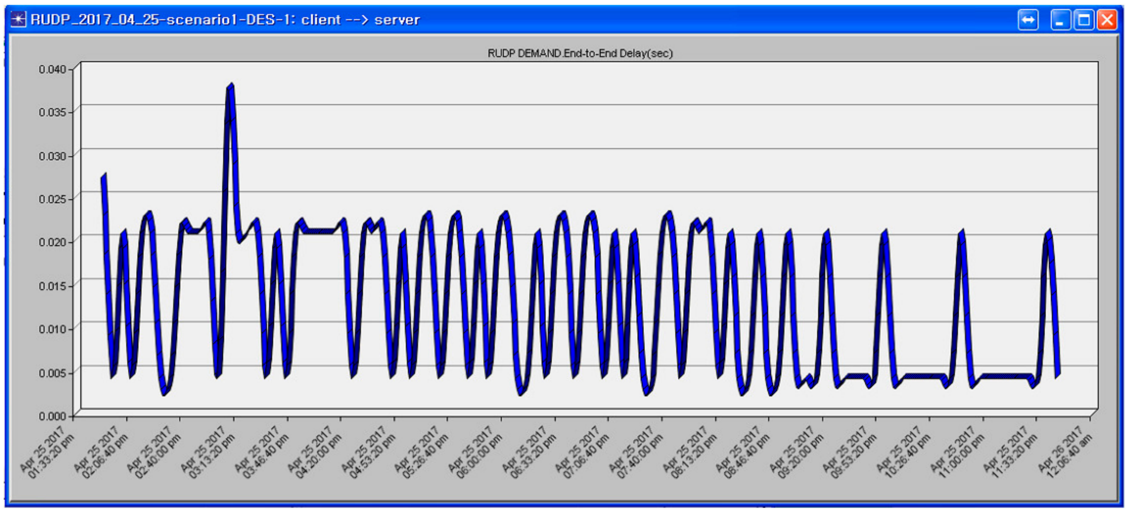

Figure 10. Results of RUDP performance evaluation scenario.

4.2. Analysis Scenario Evaluation of TCP/UDP/RUDP Scenario

The end-to-end delay in the TCP scenario showed that the delay time had increased when the packet loss situation occurred, whereas the UDP scenario showed that the delay time had remained constant. However, the packet loss was compensated to a certain degree and its delay time remained in the value range between TCP and UDP. As a result, the delay times were UDP > RUDP > TCP, starting from the lowest value.

Meanwhile, the end-to-end delay of each protocol is shown in Figures 11 and 12, respectively. There were some packet-loss situations in the UDP scenario but none for the TCP scenario as the TCP model had adopted a reliable re-transmission algorithm which activated in the event of any packet losses. The UDP model did not have such a function embedded in its algorithm so that it could not ensure 100% packet delivery. The RUDP model in terms of packet delivery performance remained between the other two models. Although the number of re-transmissions times was set at 3 in the scenario to improve the success rate of delivery, the larger number would have produced a better result, even to a 100% success rate.

The performance evaluations also revealed that the TCP model was reliable but slower whereas the UDP model was faster but lacked the reliability in packet delivery. This may be due to the characteristics of TCP which require separate connection management for each link between server and client. By contrast, the RUDP model mainly deals with the communications between clients through the server to ensure communication reliability but this process also reduces the speed as well, showing a lower speed than the speed achieved by the UDP models. Another disadvantage of the RUDP model is that the server could be overloaded when there are too many clients. In any case, RUDP is well suited for communication systems that require a steady performance in terms of both speed and reliability. Indeed, each protocol has its own merits but as this study focuses on their performances in network games, an RUDP-based communication algorithm/system is recommended.

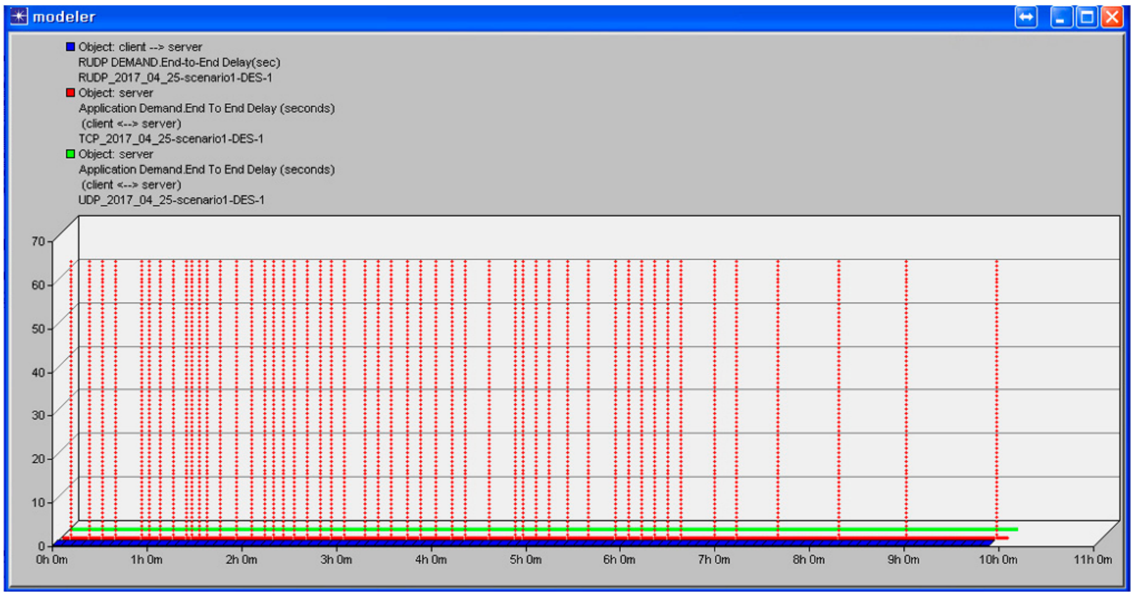

Figure 11. End-to-end delay in each protocol model (1): *X*-axis: time, *Y*-axis: end-to-end delay.

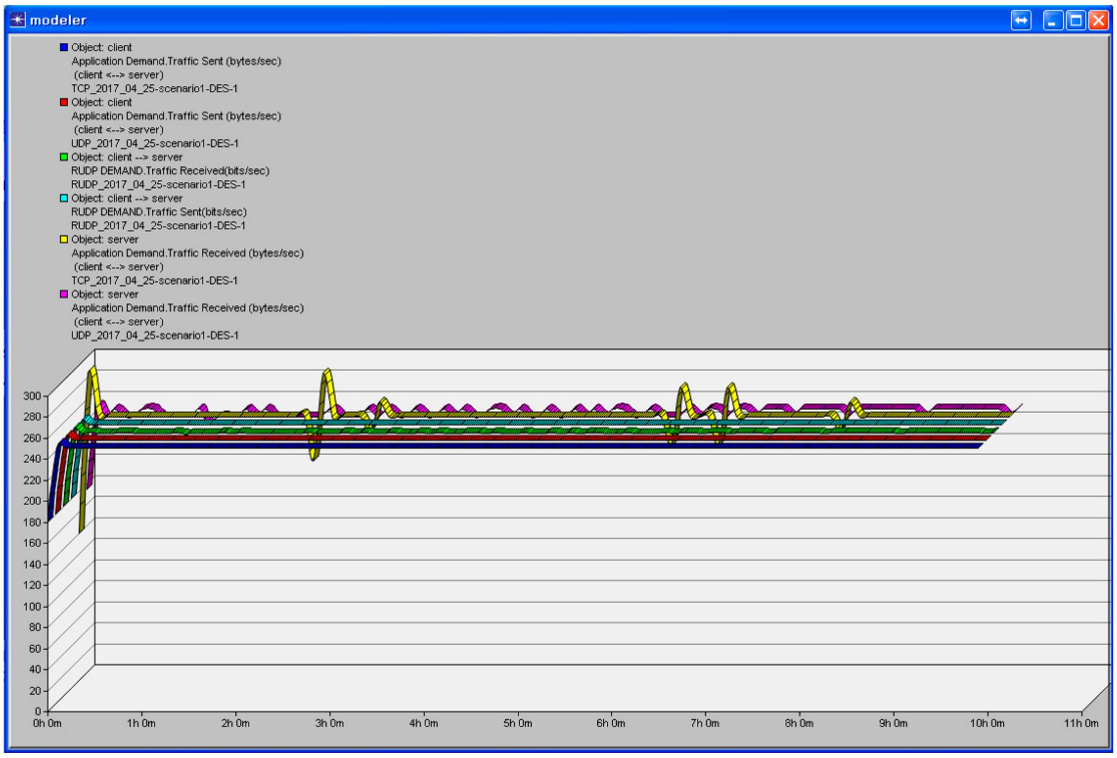

Figure 12. End-to-end delay in each protocol Model (2): *X*-axis: time, *Y*-axis: end-to-end delay.

Below, Table 2 shows the total accumulated number of received packets. TCP demonstrates a 100% success rate while RUDP and UDP demonstrate approximately 99.8% and 98.5%, respectively. Comparing the success rates based on the simulation results, RUDP had a lower success rate than TCP but higher than that of UDP. This demonstrates that RUDP is more adoptable than UDP in terms of reliability.

Table 2. Number of Received Packets.

	TCP	RUDP	UDP
Packets transmitted	5833.2	5833.2	5833.2
Packets received	5833.2	5824.8	5749.8
Success rate	100%	99.8%	98.5%

A comprehensive analysis results for the three protocols is shown in Table 3 along with their unique characteristics, which can be used to select the most suitable protocol for any communication fields.

Table 3. Comparative results.

	TCP and UDP	RUDP
Merit/Demerit	TCP is slow. UDP has low reliability.	Has both advantages.
Networking	TCP: one side has to become a server and be on standby. UDP: immediate communication is possible without being on standby.	Mainly deals with the communications between clients: the communication reliability is guaranteed by sending data via server but the speed is slower than UDP and may cause overloads to the server. RUDP compensates for packet losses to a certain degree.
Speed	TCP is higher speed. UDP is lower speed.	Faster than TCP.
Reliability	TCP: better reliability. UDP: lower reliability.	Improvement has been made to have more reliability than UDP.
Scalability	Both are being used for most networking operations.	Smartphone games. Online games.
Results	TCP is reliable but slow. UDP is fast but less reliable.	Faster than TCP.Better reliability than UDP.

5. Conclusions and Future Work

As mentioned earlier, RUDP has become the best option for network game designing that aims to solve or minimize the latency problem. The applications based on the TCP protocol offer much slower data transmission rates whereas UDP-based ones are faster but lack reliability as they have higher packet loss rates and sometimes perform an 'out-of-order' packet delivery, usually caused by the high bit errors rate (BER) or the insufficient mobility of wireless nodes. Although both TCP and UDP-based communication systems show satisfactory levels of performance in a wired network, they cannot work properly in a wireless network as they do not have some functions that allow a flawless performance in such an environment. This is mainly due to the problems in the game design process where the typical characteristics of wireless networks have not been considered. A RUDP-based model has been developed in this study to complement these problems related to latency in both environments. OPNET Modeler 14.5 PL8 was used to implement the model. Moreover, for the OPNET simulations and the performance comparisons between three protocols, re-transmission times and timeouts have been re-adjusted after each packet transmission while developing a separate application for the comparisons. As expected, TCP showed a reliable packet transmission capability, but its slow speed was not suitable for network games.

On the other hand, UDP exhibits much better speeds but its reliability was too low, meaning that it is also not the best option for current network games. As for the UDP, it has complemented these problems quite effectively but the problem of latency remains. The problem of loads in the process and memory is the result of additional complexities and remain as one of the major problems in RUDP proposed in this study. Future tasks include designing a viable congestion control mechanism to surmount such a problem. Meanwhile, in this study, the protection of the data transmitted with the protocols discussed was not considered. Therefore, data protection will be included in the author's future tasks, in addition to conducting research into the Enhanced-RUDP (E-RUDP), an improved version of RUDP. It is expected that the proposed RUDP simulation model will be useful to game developers and companies in upgrading their game design process.

The OPNET simulation model proposed in this research work for the network games is expected to be used by small or medium-sized game companies, including start-ups, for educational purposes or cost-reduction measures. The use cases will be studied further in the future.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2017R1C1B5077157).

Acknowledgments: The first part of this paper [26] was presented at the 9th KIPS International Conference on Ubiquitous Information Technologies and Applications (CUTE 2014), 17–19 December 2014, Guam, USA. The second part of this paper was presented at the Oral Session in2017 APAIS 1st Special Issue International Workshop [25], 7–8 February 2017.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Busse, M.; Lamparter, B.; Mauve, M.; Effelsberg, W. Lightweight QoS-support for networked mobile gaming. In Proceedings of the Third Workshop on Network and System Support for Games (NetGames 2004), Portland, OR, USA, 30 August 2004.
2. Mauve, M. A generic proxy system for networked computer games. In Proceedings of the First Workshop on Network and System Support for Games (NetGames 2002), Braunschweig, Germany, 16–17 April 2002; pp. 25–28.
3. Bova, T.; Krivoruchka, T. Reliable UDP Protocol, 1999. Internet-Draft. Available online: https://tools.ietf.org/id/draft-ietf-sigtran-reliable-udp-00.txt (accessed on 2 November 2018).
4. Ritter, H.; Voigt, T.; Tian, M.; Schiller, J. Experiences using a dual wireless technology infrastructure to support ad-hoc multiplayer games. In Proceedings of the Second Workshop on Network and System Support for Games (NetGames 2003), Redwood City, CA, USA, 22–23 May 2003; pp. 101–105.
5. Quax, P.; Monsieurs, P.; Lamotte, W.; De Vleeschauwer, D.; Degrande, N. Objective and subjective valuation of the influence of small amounts of delay and jitter on a recent first person shooter game. In Proceedings of the Third Workshop on Network and System Support for Games (NetGames 2004), Portland, OR, USA, 30 August 2004; pp. 152–156.
6. Mansley, K.; Scott, D.; Tse, A.; Madhavapeddy, A. Feedback, latency, accuracy: Exploring trade-offs in location-aware gaming. In Proceedings of the Third Workshop on Network and System Support for Games (NetGames 2004), Portland, OR, USA, 30 August 2004; pp. 93–97.
7. Huh, J.H.; Seo, K. An Indoor Location-Based Control System Using Bluetooth Beacons for IoT Systems. *Sensors* **2017**, *17*, 2917. [CrossRef] [PubMed]
8. Fritsch, T.; Ritter, H.; Schiller, J. The effect of latency and network limitations on MMORPGs—A Field Study of Everquest 2. In Proceedings of the Fifth Workshop on Network and System Support for Games (NetGames 2005), Hawthorne, NY, USA, 10–11 October 2005.
9. Yasui, T.; Ishibashi, Y.; Ikedo, T. Influence of Network Latency and Packet Loss on Consistency in Networked Racing Games. In Proceedings of the Fifth Workshop on Network and System Support for Games (NetGames 2005), Hawthorne, NY, USA, 10–11 October 2005.
10. Evolved Universal Terrestrial Radio. Radio Resource Control (RRC) Protocol Specification, 3GPP Tech. Spec. 25.331, v5.1.0, 2002. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2440 (accessed on 2 November 2018).
11. Petrak, L.; Landsiedel, O.; Wehrle, K. Framework for evaluation of networked mobile games. In Proceedings of the Fifth Workshop on Network and System Support for Games (NetGames 2005), Hawthorne, NY, USA, 10–11 October 2005.
12. Janecek, A.; Hlavacs, H. Programming Interactive Real-Time Games over WLAN for Pocket PCs with J2ME and NET CF. In Proceedings of the Fifth Workshop on Network and System Support for Games (NetGames 2005), Hawthorne, NY, USA, 10–11 October 2005.

13. Tan, S.A.; Lau, W.; Loh, A. Networked game mobility model for first person-shooter games. In Proceedings of the Fifth Workshop on Network and System Support for Games (NetGames 2005), Hawthorne, NY, USA, 10–11 October 2005.
14. Beigbeder, T.; Coughlan, R.; Lusher, C.; Plunkett, J.; Agu, E.; Claypool, M. The effects of loss and latency on user performance in unreal tournament. In Proceedings of the Third Workshop on Network and System Support for Games (NetGames 2004), Portland, OR, USA, 30 August 2004; pp. 144–151.
15. Pellegrino, J.; Dovrolis, C. Bandwidth requirement and state consistency in three multiplayer game architectures. In Proceedings of the Second Workshop on Network and System Support for Games (NetGames 2003), Redwood City, CA, USA, 22–23 May 2003.
16. Liu, Y.; Wang, J.; Kwok, M.; Toulouse, J.D.M. Capability of IEEE 802.11g Networks in Supporting Multi-player Online Games. In Proceedings of the Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 8–10 January 2006.
17. Cheung, G.; Sakamoto, T.; Sweeney, M. Performance enhancing proxy for interactive 3G network gaming. In Proceedings of the Second International Symposium on Multimedia over Wireless (ISMW), Vancouver, BC, Canada, 3–6 July 2006.
18. Huh, J.-H.; Je, S.-M.; Seo, K. Communications-based technology for smart grid test bed using OPNET simulations. In Proceedings of the ICISA 2016, Ho Chi Minh City, Vietnam, 15–18 February 2016; Springer: Singapore, 2016; pp. 227–233.
19. Smed, J.; Kaukoranta, T.; Hakonen, H. *A Review on Networking and Multiplayer Computer Games*; Technical Report 454; Turku Centre for Computer Science: Turku, Finland, 2002.
20. SNAP Communications: Client Developers' Guide. Available online: https://www.sega.com/ (accessed on 1 September 2018).
21. Akkawi, A.; Schaller, S.; Wellnitz, O.; Wolf, L. A mobile gaming platform for the IMS. In Proceedings of the Third Workshop on Network and System Support for Games (NetGames 2004), Portland, OR, USA, 30 August 2004.
22. Farber, J. Network game traffic modeling. In Proceedings of the First Workshop on Network and System Support for Games (NetGames 2002), Braunschweig, Germany, 16–17 April 2002; pp. 53–57.
23. Borella, M.S. Source models of network gaming traffic. In *Computer Communications*; Elsevier: Amsterdam, The Netherlands, 2000; pp. 403–410.
24. Ghosh, P.; Basu, K.; Das, S.K. Improving end-to-end quality-of-service in online multi-player wireless gaming networks. In *Computer Communications*; Elsevier: Amsterdam, The Netherlands, 2008; Volume 31, pp. 2685–2698.
25. Huh, J.-H. A Study on the Application of Reliable User Datagram Protocol to Handle Latencies in Network Games. In *Asia-Pacific Proceedings of Applied Science and Engineering for Better Human Life*; APAIS Publishing: Tasmania, Australia, 2017; Volume 12, pp. 57–61.
26. Huh, J.-H.; Seo, K. RUDP Design and Implementation Using OPNET Simulation. In *Computer Science and its Applications*; CUTE 2014 at Guam, Lecture Notes in Electrical Engineering; Springer: Berlin/Heidelberg, Germany, 2015; Volume 330, pp. 913–919.
27. What is Going on Behind a Net Game? From the Perspective of Network Engineers, the Principle of Game Design. Available online: http://www.4gamer.net/games/105/G010549/20100905002/ (accessed on 1 September 2018). (In Japanese)
28. Haris, K.; Oskar, K. Applying Automated Testing in an Existing Client-Server Game: A Pursuit for Fault Localization in Quake 3. Bachelor's Thesis, Linnaeus University, Linnaeus, Sweden, 2015; pp. 1–18.
29. Huh, J.-H.; Seo, K. RUDP Design and Implementation using OPNET. In Proceedings of the 2014 Korea Computer Congress (KCC), Jeju, Korea, 29 June–1 July 2014; pp. 1–3.
30. Abdullah Al Mamun, M.; Rahman, M.; Tan, H. Performance Evaluation of TCP over Routing Protocols for Mobile Ad Hoc Networks. In Proceedings of the First International Conference on Communications and Networking in China, Beijing, China, 25–27 October 2006; pp. 1–3.
31. Xylomenos, G.; Polyzos, G. TCP and UDP Performance over a Wireless LAN. In Proceedings of the INFOCOM'99: Conference on Computer Communications, New York, NY, USA, 21–25 March 1999; Volume 2, pp. 439–446.
32. Velten, D.; Hinden, R.; Sax, J. Reliable Data Protocol. Available online: https://tools.ietf.org/html/rfc908 (accessed on 2 November 2018).

33. Gehlen, G.; Aijaz, F.; Walke, B. Mobile Web Service Communication over UDP. In Proceedings of the IEEE 64th Vehicular Technology Conference, Montreal, QC, Canada, 25–28 September 2006; pp. 1–5.
34. Le, T.; Kuthethoor, G.; Hansupichon, C.; Sesha, P.; Strohm, J.; Hadynski, G.; Kiwior, D.; Parker, D. Reliable User Datagram Protocol for airborne network. In Proceedings of the Military Communications Conference, (MILCOM 2009), Boston, MA, USA, 18–21 October 2009; pp. 1–6.
35. Thammadi, A. Reliable User Datagram Protocol (RUDP). Master's Thesis, Kansas State University, Manhattan, KS, USA, 2011; pp. 1–30.
36. Shim, K.H.; Park, I.K.; Chung, J.; Lee, E.H.; Kim, J.S.; Choi, B.T. Distance Based Distributed Online Game Server System. U.S. Patent 20040116186A1, 17 June 2004.
37. Masirap, M.; Amaran, M.H.; Yussoff, Y.M.; Ab Rahman, R.; Hashim, H. Evaluation of reliable UDP-based transport protocols for Internet of Things (IoT). In Proceedings of the IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Batu Feringghi, Malaysia, 30–31 May 2016.