

An Enhanced Method for Mitigation of Network Traffic using TCP Signalling Control

Won Seok Choi*, Seong Gon Choi*

*College of Electrical & Computer Engineering, Chungbuk National University, South Korea

wschoi@cbnu.ac.kr, sgchoi@cbnu.ac.kr

Abstract— This paper proposes an enhanced method for mitigation of network traffic using TCP packet control in a GGSN (Gateway GPRS Support Node). In proposed method, the GGSN determines whether the received packet is the TCP signalling or not according to the information of packet size, and the TCP signalling holding time adjusted by network traffic load status. Through the OPNET simulation, we demonstrate that the network traffic load can be reduced.

Keywords— Network Traffic Load, TCP Signalling Algorithm, GGSN, OPNET simulation

I. INTRODUCTION

Nowadays, the mobile communication user is rapidly growing with an increase of mobile communication application market. Accordingly, the demand for smartphone, which supports mobile internet data service as well as typical voice call, is sharply increasing. In fact, smartphones have been experiencing a sensational evolution over the past year, starting from simple devices with only voice call services to smartphones offering novel services such as Social Network Service (SNS, e.g. Kakao-talk, Google-talk, Facebook, etc.), multimedia streaming service (e.g. Video, VoIP, etc.), web surfing, gaming, etc. [1].

The increase of smartphone is rapidly increasing the network traffic load as well as network equipment overload. Especially, SNS is one of the most popular services for smartphone in mobile application market. Also, the user uses the multimedia streaming service as real time service via smartphone. As a result, both SNS and multimedia streaming service seriously impacts on network and network equipment [2].

Although SNS application generates the small-sized packets, the network or network equipment traffic load increases from tremendous service requests such as text message, signaling message, keep-alive message, etc. In fact, the keep-alive messages are one of the most cause results in the network equipment overload. Because the keep-alive message is generated and sent to the agent server even though the smartphone is in idle mode [3]. This result in many additional works of the network equipment of the GGSN (Gateway GPRS Support Node).

On the other hand, multimedia streaming service generates the large-sized packets. Though the request frequency of

multimedia traffic is little, the network or network equipment traffic load increase according to the large traffic size. Consequently, when the traffic load of the network or the network equipment is overloaded, the various problems are caused such as application server down [4], bottleneck phenomenon [5], decrease of the QoS (quality of service) [6], etc.

To solve the network or network equipment overload, the network provider tries to install the additional network equipment (e.g. Content Delivery System, Cloud Computing), and the service provider installs the related server (e.g. push server) [7] and extends the period of the keep-alive message. However, these are not fundamental solution to reduce network or network equipment traffic load.

Therefore, for solving the network or network equipment overload, we propose a TCP signaling delay algorithm which operates on the GGSN. By using proposed method, the delay time is varied according to the packet size (e.g. packet size of TCP SYN) and the traffic load status. In proposed method, the network or network equipment overload can be effectively reduced because the following data packets are also delayed in case the TCP signaling packets are delayed.

In order to analyze the performance of proposed method, we use the OPNET simulation tool. As a result, we demonstrate that the network load can be reduced when the TCP signaling is delayed.

This paper is structured as follows. Section 2 describes the related work. In section 3, we present network traffic load reduction method using TCP signaling delay algorithm. In section 4, we show the performance analysis using OPNET simulation. Finally, we conclude in section 5.

II. RELATED WORK

A. Traffic shaping, traffic policing, and traffic priority control

The existing studies for mitigating network traffic load are traffic shaping, traffic policing, and traffic priority control method. The traffic shaping method provides the solution for unexpected traffic like the burst traffic by normalizing average rate of the data transmission. When the traffic is transmitted by the traffic shaping, the traffic policing is method which observes corresponding traffic flows. It observes whether SLA (Service Level Agreement) between the user and the network

is achieved. The traffic priority control is method which determines the priority between the real-time traffic and the non-real-time traffic and improves the service quality by processing the corresponding traffic on the queue.

In the 3G radio access network, the framework of traffic conditioning with QoS provisioning has been proposed. In this paper, the main idea of our traffic conditioning approach is to employ traffic shaping at each user equipment (UE) and traffic policing at the RNC. The traffic generated at each UE is regulated by a traffic shaper in the form of a token bucket, and the conformance of the traffic is policed at the RNC according to traffic policing policies [8].

However, when the traffic load of the network or the network equipment is overloaded, the application of the traffic shaping and the traffic policing for reducing the traffic load is actually difficult since the traffic shaping drops much packets for processing fast-growing traffic. This results in the problems like the service suspension. Also, in traffic policing, when the traffic load of the network is overloaded, the SLA between user and network is not performed.

The comparing effect of traffic shaping and traffic policing on aggregate traffic dynamics method has been proposed. Especially stochastic properties on traffic time series. This paper simulates a series of tests and compares the result obtained from OPNET simulation program to simulation with synthetic traffic generated by chaotic dynamic systems [9].

However, this paper only interested in parameters such as Hurst parameter, autocorrelation function and variance for different shaping and policing techniques applied to input traffic in order to obtain smooth aggregate traffic trace. In addition, the paper mentioned above just experiments the synthetic traffic generated by simulator except for the real world traffic and the method reducing the network load is not considered.

The Home Gateway Energy Saving Mechanism (HGESM) which operates in the home gateway has been proposed. In this paper, a HG determines whether the received packet is the real time or non-real time. And then, the non-real time packet is adjusted to according to the queue threshold and timer in a HG [10].

However, the traffic priority control has the problem that it provides one-sided service quality regarding the real-time traffic.

B. CDN(Content Delivery Network)

A content delivery network is a large distributed system of servers deployed in multiple data centers in the internet. The goal of a CDN is to serve content to end users with high performance and network load distribution. CDNs act as trusted overlay networks that offer high-performance delivery of common Web objects, static data, and rich multimedia content by distributing content load among servers that are close to end users. Accordingly, researchers have widely considered CDNs to be an effective solution to reducing network traffic load and high-performance [11].

However, it is impossible that CDN handles all of the applications traffic generated from user. In addition, building

CDNs expensive cost as much as constructing an additional access network.

III. PROPOSED METHOD

In this section, we describe a TCP signaling delay algorithm which operates on the GGSN for reducing network traffic load. Before describing the proposed method, we investigate the signaling procedures (e.g. TCP SYN, FIN, etc.) and the features (e.g. size of TCP packet) of various applications on the smartphone.

For this purpose, we tested by using a Samsung Galaxy-Tab and a wireshark tool for analyzing the packet size and the features of mobile data flows. Here, we select the wireshark tool among network analyzers because the wireshark provides a useful functions such as easy-operation and available on wireless environment, especially on smartphone, etc.



Figure 1. A method on analyzing application traffic using wireshark tool in a Galaxy-Tab.

Figure 1 shows the method on analyzing application traffic using wireshark in a Galaxy-Tab. First of all, we select the network used in the experiment. Here, we chose the 3G network (UMTS) because of the 3G network used to the network overload. And then, we run the wireshark application. While the wireshark application is running, we execute the application related to our experiments such as SNS, multimedia, etc. Next, when the experiment finished using application, we analyze the captured packets by wireshark in the PC.

A. The analysis of various applications

Figure 2 shows the captured packet of Kakao-talk. According to the experimental result, Kakao-talk mutually sends and received many packets for transmitting a message between a user and corresponding server using HTTP and TCP packets. In fact, most of the SNS applications show similar as the Figure 2 because HTTP and TCP widely used in non-real-time service.

Also, we can confirm the information such as used protocol, packet length and the procedures of TCP three-way handshaking. Among this information, the most important things that the TCP initialize packet (e.g. TCP [SYN], TCP [SYN, ACK], and TCP [ACK], etc.) size is less than 68bytes. The different SNS applications have same features as figured 2.

No.	Time	Source	Destination	Protocol	Length	Info
247	3013.03000	211.234.229.23	10.223.123.224	DNS	290	standard query response A 203.240.
248	5613.08505	10.223.123.224	203.246.172.36	TCP	68	59359 > http [SYN] Seq=0 Win=65535
249	5613.09378	110.76.140.33	10.223.123.224	TCP	68	http > 33646 [SYN, ACK] Seq=0 Ack=
250	5613.09399	10.223.123.224	110.76.140.33	TCP	56	33646 > http [ACK] Seq=1 Ack=1 Win
251	5613.10310	10.223.123.224	110.76.140.33	HTTP	540	POST /android/friends/update.json
252	5613.21478	203.246.172.36	10.223.123.224	TCP	68	http > 59359 [SYN, ACK] Seq=0 Ack=
253	5613.21495	10.223.123.224	203.246.172.36	TCP	56	59359 > http [ACK] Seq=1 Ack=1 Win
254	5613.26407	110.76.140.33	10.223.123.224	TCP	56	http > 33646 [ACK] Seq=1 Ack=485 W
255	5613.34539	10.223.123.224	203.246.172.36	HTTP	430	GET /android/account/settings.json
256	5613.48384	110.76.140.33	10.223.123.224	TCP	1436	[TCP segment of a reassembled PDU]
257	5613.48398	10.223.123.224	110.76.140.33	TCP	56	33646 > http [ACK] Seq=485 Ack=138
258	5613.52386	110.76.140.33	10.223.123.224	TCP	1436	[TCP segment of a reassembled PDU]
259	5613.52398	10.223.123.224	110.76.140.33	TCP	56	33646 > http [ACK] Seq=485 Ack=276
260	5613.56393	110.76.140.33	10.223.123.224	TCP	1436	[TCP segment of a reassembled PDU]
261	5613.56412	10.223.123.224	110.76.140.33	TCP	56	33646 > http [ACK] Seq=485 Ack=414
262	5613.60457	110.76.140.33	10.223.123.224	TCP	1436	[TCP segment of a reassembled PDU]

Figure 2. The captured packet of Kakao-talk

Figure 3 describes the keep-alive packet of yahoo-messenger. In figure 3, the background traffic exists and flows even if the smartphone is in idle mode. Therefore, the number of request traffic is transmitted to corresponding server or network equipment (e.g. GGSN). If tremendous requests transmit to the network generated from users, it will seriously impact on the network load and network equipment load. Actually, this phenomenon has taken place in network providers [12].

209.73.35.134	204.236.220.251	216.155.194.30	216.155.195.253
45696 > http [SYN]			
http > 45696 [SYN]			
45696 > http [ACK]			
POST /v1/session/ke			
HTTP/1.1 200 OK 0			
http > 45696 [FIN]			
45696 > http [ACK]			
HTTP/1.1 200 OK (a			
33343 > http [ACK]			
45696 > http [ACK]			
GET /v1/pushchannel			
http > 33343 [ACK]			
45696 > http [FIN]			
http > 45696 [ACK]			
[TCP Keep-Alive] ht			
[TCP Keep-Alive ACK			
[TCP Keep-Alive ht			

Figure 3. The background traffic of yahoo-messenger at smartphone in idle mode [12].

For obtaining more certain result, we analyzed various applications, and summarized the results in TABLE I .

As shown the TABLE I , we can see that the number of the applications uses the three-way handshaking of TCP packets for accessing to the application server. Also, the signalling packet (e.g., SYN, ACK, FIN) size of TCP is less than 68bytes. On the other hand, the PDU and retransmission packets have the various packet sizes from 93bytes to 1436bytes. As a result, this feature can be used in bottleneck point (e.g., Edge Router, GGSN) through the fast classification of the packets.

The different application (e.g., Yahoo-messenger, YouTube, Twitter, Facebook, Google, etc.) not mentioned in TABLE I has same features as presented in TABLE I .

TABLE 1. the summary of application features [2]

Name	Features of Applications		
	Message	TCP Type	Size of Packet(byte)
Kakao-Talk My-people Naver-Talk, Google-Talk etc.	Text Picture Voice Video Emoticon	SYN	56~68
		SYN, ACK	56~68
		ACK	56~68
		FIN, ACK	56~68
		Dup ACK	56~68
		RST	56
		Segment of a reassembled PDU	1392 ~ 1436
		Retransmission	93 ~ 1436

B. An enhanced method for mitigation network traffic

In previous section A, from the analysis, we can see that the TCP signaling packet (e.g. TCP [SYN], TCP [SYN, ACK], TCP [ACK], etc.) of TCP has constant size(less than 68bytes). Therefore, we regard the TCP signaling packet which is less than Threshold 1(that is, TCP signaling packet or 68bytes) among the TCP packets.

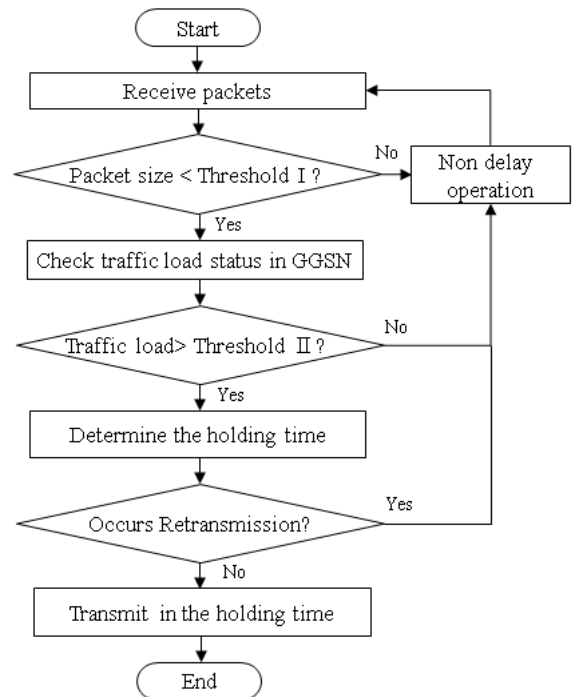


Figure 4. The flow chart of the TCP signaling delay algorithm which operates on the GGSN

Figure 4 presents the flow chart of the TCP signaling delay algorithm which operates on the GGSN for reducing network traffic load or network equipment load. The GGSN has the TCP signaling delay algorithm function. The algorithm checks only the frame size of MAC header for verifying the TCP signaling packet. Checking only MAC header enhanced processing time in a GGSN because the algorithm does not

parsing other headers such as IP header, TCP header and etc. [13].

In figure 4, First of all, the GGSN receives the packets. If the packets total size of received packets is larger than Threshold 1, the GGSN operates to the non-delay operation. Non-delay operation means that the TCP signaling delay algorithm function does not be supported. On the other hand, if the total packet size of received packet is less than Threshold 1, the GGSN checks the Traffic load status (e.g. CPU utilization, Queue utilization, etc.) itself. If the traffic loads less than Threshold 2 (that is, the remainder of the queue size of the whole queue size, e.g. 50%), the GGSN operates to the Non-delay operations. On the other hand, if the traffic loads larger than Threshold 2, the GGSN determines the holding time with TCP signaling packet.

At this time, the holding time of the TCP signaling packet is calculated within the time which the retransmission is not occurred. And then, if the retransmission occurs in a GGSN with a TCP signaling packet, the GGSN works non-delay operation because of the retransmission packets result in the additional network load. On the other hand, if the retransmission does not occur, the GGSN holds the TCP signaling packet. After that, the GGSN transmits the TCP signaling packet in the holding time.

Accordingly, the network traffic load or network equipment load can be effectively reduced because the following data packets are also delayed in case the TCP signaling packets are delayed.

IV. PERFORMANCE ANALYSIS

This section presents the performance analysis for the proposed method. For analyzing the TCP signaling delay algorithm, we use the OPNET simulation, and simulate the various values (e.g. CPU utilization, Forwarding Memory Queue Size, etc.) according to holding time (s) and arrival rate (Kbytes) on the GGSN.

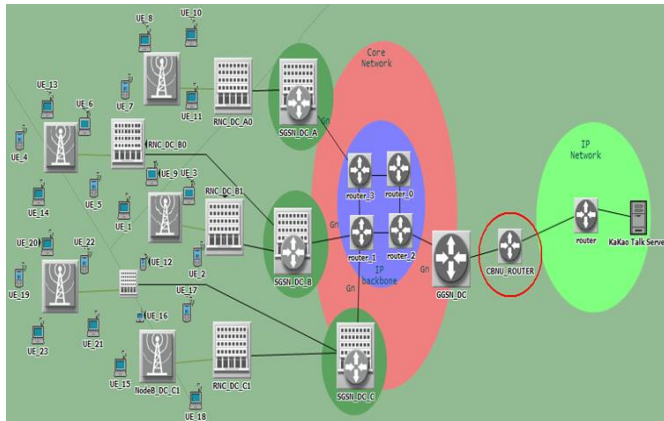


Figure 5. Network configuration of the proposed method

Figure 5 shows the network configuration (UMTS, Universal Mobile Telecommunications System) of the proposed method, and the TCP signaling delay algorithm was implemented on the CBNU_ROUTER. Concretely, the CBNU_ROUTER checks the frame size of MAC header whether the checked packet is the TCP signaling or not. Next,

the CBNU_ROUTER holds the TCP signaling packet with some time (e.g. 0.2second). And then, the CBNU_ROUTER transmits the TCP signaling packet after holding time.

For the convenience of simulation, the CBNU_ROUTER was used the same GGSN_DC, and the CBNU_ROUTER acts as a gateway between UMTS and Internet.

TABLE 2. THE SIMULATION PARAMETERS IN FIGURE 4

The simulation parameters		
Name	Value	State
Number of UE	23	Fixed
Packet type of UE	HTTP and TCP	Fixed
HTTP and TCP packet size	64bytes ~ 1500bytes	Variable
Traffic size(per-UE)	4Kbytes ~ 40Kbytes	Variable
Inter-request time(per-UE)	10ms	Fixed
Arrival rate(λ) in a CBNU_ROUTER	4Mbt/s ~ 1Gbytes	Variable
Service rate(μ) in a CBNU_ROUTER	1Gbps	fixed
Holding time in a CBNU_ROUTER	0s ~ 0.8s	Variable

Table II presents the simulation parameters in figure 5. We set the parameters into the user-equipment and the CBNU_ROUTER for simulation. After that, we change the parameters both the traffic size (per-UE) in a UE and the holding time in a CBNU_ROUTER while the simulation operates.

The state values are divided into “fixed” and “available”. The fixed value does not effect by external conditions (e.g. Traffic size, Arrival rate, holding time). For example, the number of UE values is “fixed” while simulation is performed. On the other hand, the available value effect by external conditions. For example, the traffic size and the holding time values are changed by simulator. Also, the arrival rate of the CBNU_ROUTER value is “available” since it is determined by the traffic size by generated from the per-UE.

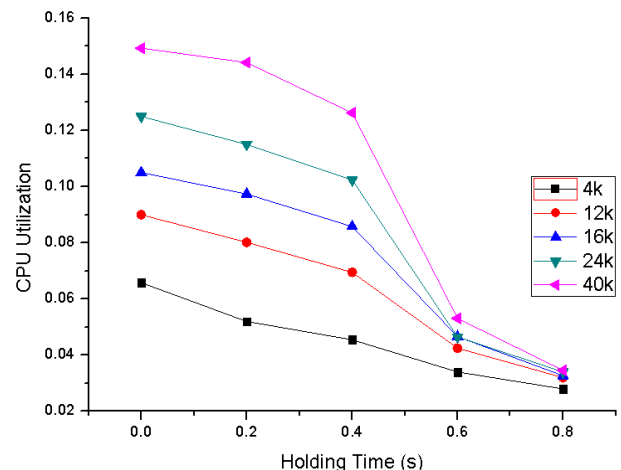


Figure 6. The CPU utilization according to the holding time in a CBNU_ROUTER.

Figure 6 shows the average CPU utilization according to the holding time in a CBNU_ROUTER. The TCP signaling packets that are generated by the user was applied to the holding time in a CBNU_ROUTER. As a result, as the holding time of TCP signaling packet is increasing, the average CPU utilization is decreased. In other words, this means that the network traffic is saved on the buffer of users because the service initiation is delayed.

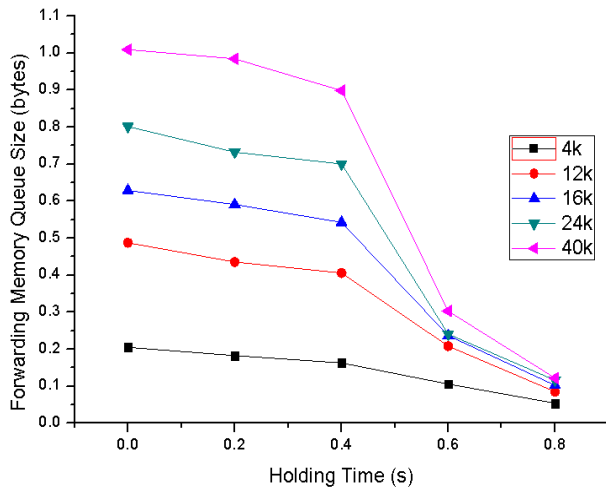


Figure 7. Forwarding memory Queue Size according to holding time in a CBNU_ROUTER.

Figure 7 presents the average forwarding Memory Queue Size (bytes) according to the holding time in a CBNU_ROUTER. As the holding time increasing, we can be seen that the forwarding memory queue size is decreased. Here, the forwarding memory queue size means that the received packet is saved to queue on the CBNU_ROUTER.

Through the above figures, we can identify that the network equipment overload effectively reduced by using TCP signaling delay algorithm. Also, if the network equipment overload is reduced, the network overload is naturally mitigated.

Consequently, if the TCP signaling packets are holding in the GGSN, the network load is mitigated because the network load moved to the user devices.

V. CONCLUSION

This paper proposes a TCP signaling delay algorithm which operates on the GGSN for reducing the network traffic load. In proposed method, when the network is overloaded, the TCP signaling response time is adjusted to according to network traffic load status. For the performance analysis, we use OPNET simulation tool. As a result, the proposed method shows that the network traffic load can be effectively reduced by using our algorithm. Therefore, the proposed method can be applied to the bottleneck point which processes the TCP signaling packets such as GGSN, SGSN, etc.

ACKNOWLEDGMENT

This work was partly supported by the knowledge economy technical innovation business program of MKE [10043135, Establishment of energy-efficient infrastructure on IT network systems]

*Corresponding Author : Seong Gon Choi (sgchoi@cbnu.ac.kr)

REFERENCES

- [1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015," G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (references Cisco, February 2011).
- [2] Min Woo Kim, Dong Geun Yun, Jong Min Lee, Young Jae Shim, Seong Gon Choi, "Network Traffic Mitigation Method Using TCP Signalling Delay Algorithm," Advanced Communication Technology (ICACT), 2012 14th International Conference on, pp.730-733, February 2012.
- [3] Hyeon Yang, Min Woo Kim, Jong Min Lee, Seong Gon Choi, "Battery Life Time Extension Method by using Signaling Interval Control," Advanced Communication Technology (ICACT), 2012 14th International Conference on, pp.327-330, February 2012.
- [4] Min Woo Kim, Seong Gon Choi, "Video Quality Improvement System by using Meta-Signalling and Server Switichg Mechanism in Hierarchical Architecture," ICCE2012 Submitted for publication.
- [5] Jong Min Lee, Hyo-Jin Park, Seong Gon Choi, Jun Kyun Choi, "Adaptive Hybrid Transmission Mechanism for On-Demand Mobile IPTV Over WiMAX," Broadcasting, IEEE Transactions on, vol.55, no.2, pp.468-477, June 2009.
- [6] Hyun Jong Kim, Seong Gon Choi, "Design of Video Streaming Service Quality Control System using Available Bandwidth Information," Advanced Communication Technology (ICACT), 2011 13th International Conference on, pp.1540-1544, February 2011.
- [7] Vakali, A., Pallis, G., "Content Delivery Network: Status and Trends," Internet Computing, IEEE, vol.7, no.6, pp.68-74, November-December 2003.
- [8] Frank Yong Li, Stol, N., "QoS Provisioning using Traffic Shaping and Policing in 3rd-generation Wireless Networks," Wireless Communications and Networking Conference (WCNC), 2002 IEEE, vol.1, pp.139-143, March.2002.
- [9] Daian, D.-S., Giura, D.-H., "Traffic Shaping and Traffic Policing Impacts on Aggregate Traffic Behaviour in High Speed Networks," Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on, pp.465-467, May.2011.
- [10] Hyeon Yang, Myong Ju Yu, Dong Geun Yun, Seong Gon Choi, "Energy Saving Mechanism using Sleep Mode of Processor in Home Gateway," Consumer Electronics (ICCE), 2012 IEEE International Conference on, pp.704-705, January 2012.
- [11] Fortino, G., Russo, W., Mastroianni, C., Palau, C.E., Esteve, M., "CDN-Supported Collaborative Media Streaming Control," Multimedia, IEEE, vol.14, no.2, pp.60-71, April-June 2007.
- [12] Min Woo Kim, Dong Geun Yun, Jong Min Lee, Seong Gon Choi, "Battery Life Time Extension Method using Selective Data Reception on Smartphone," Information Networking (ICOIN), 2012 International Conference on, pp.468-471, February 2012.
- [13] Min Woo Kim, Dong Geun Yun, Jong Min Lee, Young Jae Shim, Seong Gon Choi, "Network Traffic Mitigation Method Using TCP Signalling Delay Algorithm," Advanced Communication Technology (ICACT), 2012 14th International Conference on, pp.730-733, February 2012.