International Journal of Internet, Broadcasting and Communication Vol.11 No.3 8-19 (2019)
http://dx.doi.org/10.7236/IJIBC.2019.11.3.8

IJIBC 19-3-2

A Comparative Study on 3D Data Performance in Mobile Web Browsers in 4G and 5G Environments

Duckkyoun Nam[1], Daehyeon Lee[2], Seunghyun Lee[3], Soonchul Kwon[2*]

[1]*Department of Holography 3D Contents, Kwangwoon University, Korea*
[2*]*Graduate School of Smart Convergence, Kwangwoon University, Korea*
[3]*Ingenium College, Kwangwoon University, Seoul, Korea*
*dknam@sk.com, {dleogus13813, shlee, *ksc0226}@kw.ac.kr*

Abstract

Since their emergence in 2007, smart phones have advanced up to the point that 5G mobile communication in 2019 started to be commercialized. Accordingly, now it is possible to share 3D modeling files and collaborate by means of a mobile web. As the recently commercialized 5G mobile communication network is so useful in sharing 3D modeling files and collaborating that even large-size geometry files can be transmitted at ultra high speed with ultra low transfer delay. We examines characteristics of major 3D file formats such as STL, OBJ, FBX, and glTF and compares the existing 4G LTE (Long Term Evolution) network with the 5G NR (New Radio) mobile communication network. The loading time and packets of each format were measured depending on the mobile web browser environments. We shows that in comparison with 4G LTE, the loading time of STL and OBJ file formats were reduced as much as 6.55 sec and 9.41 sec, respectively in the 5G NR and Chrome browsers. The glTF file format showed the most efficient performance in all of the 4G/5G mobile communication networks, Chrome, and Edge browsers. In the case of STL and OBJ, the traffic was relatively excessive in 5G NR and Edge browsers. The findings of this study are expected to be utilized to develop a 3D file format that reduces the loading time in a mobile web environment.

Keywords: *3D Modeling, glTF, Comparison, Open GL, Mobile WebGL, 4G LTE, 5G NR*

1. Introduction

Since 5G mobile communication service was commercialized in Korea on April 3, 2019, for the first time around the globe, the competition of securing AR/VR-based realistic contents has been accelerated. It has become possible to operate 3D objects in a mobile web by means of the ultra-wide band/ultra-low delay transmission technology, which is a major feature of the 5G mobile communication network. Common formats of 3D modeling files include STL (STereoLithography), OBJ (Wavefront file format specification), FBX (Filmbox), and glTF (open GL Transmission Format). File sizes and characteristics are varied depending on each 3D file format since the development purposes are different [1-3]. Accordingly, the present study aims to

Manuscript Received: May. 9, 2019 / Revised: May. 14, 2019 / Accepted: May. 17, 2019
Corresponding Author: ksc0226@kw.ac.kr
Tel: +82-2-940-8637, Fax: +82-50-4174-3258
Graduate School of Smart Convergence, Kwangwoon University, Korea

compare their loading performances in a mobile web.

This paper consists of the following sections: Chapter 2 presents a general overview of the mobile communication network and compares transmission rates of each generation. Additionally, the performance of the WebGL in a mobile setting is examined. Chapter 3 includes an experiment where the 3D model format loading time is compared between the 4G and 5G mobile network communication networks. Chapter 4 comprehensively analyzes the difference in loading rates among these formats. Chapter 5 presents conclusions based on experiment results as well as future research plans.

2. Background Theory

2.1 Overview and Transmission Rate of the 5G Mobile Communication Network

A mobile communication system is designed to make it possible for users to make voice calls and utilize data freely anywhere by means of a personal terminal. Since the first generation analogue AMPS (Advanced Mobile Phone System) for voice calls emerged in Korea in the year of 1984, this technology has continued advancing up to the point that the 5G mobile communication system was finally commercialized in 2019 to realize IoT, AI, VR, and so forth. In 1996, the 2nd generation CDMA (Code Division Multiple Access) adopting the digital mode for the first time started to be commercialized in Korea for the first time around the globe. This made possible not only voice calls but also text messaging and e-mailing. From the 3G WCDMA (Wideband Code Division Multiple access) onward, the IMT-2000 international standard was adopted as an asynchronous European mode, making possible video calls and Internet access. Smart phones such as Apple iPhone and Samsung Galaxy became popular thereafter. Figure 1 shows the progress of downlink transmission rates after smartphones were first released. As 4G LTE (long-term evolution) emerged, the transmission rate on a network became an important factor. On April 3, 2019 in Korea, 5G NR (New Radio) was commercialized for the first time in the world. this technology goes beyond the boundary of mobile phones and connects every electronic device, drawing keen attention from the world as a base technology of the 4th Industrial Revolution[4].

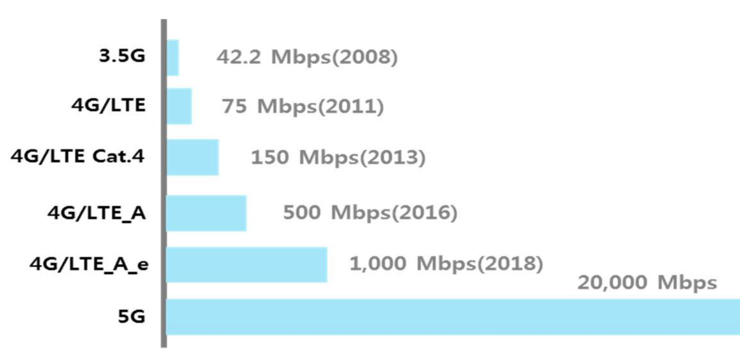


Figure 1. Downlink speed by technical generation

The competition for high transmission rates in a wireless section is viewed as having started with the 4G technology. Different 5 frequency bands (20MHz) are grouped and used simultaneously. As the bandwidth is broader, it is possible to secure a transmission rate up to 1Gbps. Based on such high transmission rates, a variety of multimedia services are provided such as video transmission and web-casting service. Figure 2 compares the performances of 4G and 5G mobile communication systems. In the 5G setting, physical resources are virtualized over the entire sections of a mobile telephone system (wireless-transmission-exchange), voice calls and data communication are integrated, and control signals are separated from user signals for flexible resource distribution. In this manner, ultra-low delay and hyper-connectivity, and ultra-high speed services are secured[5].

4G_LTE		vs.	5G_NR	
social media, music and video streaming, etc.	10ms	Latency	<1ms	Autonomous vehicles, Remote surgery, etc.
Smart utility meters, Vehicles tracking, etc.	100 Thousand Connections/km2	Connection Density	1 Million Connections/km2	Smart cities, Smart factories, etc.
Mobile video, social media, internet and game	1Gbps	Peak Data Rate	20Gbps	Mobile 4K video and game streaming, AR and VR

Figure 2. Comparing 4G and 5G

2.2 WebGL Graphics

Figure 3 shows the procedures of processing various types of 3D data stored in a web server by means of WebGL 1.0 (OpenGL ES 2.0) installed in a mobile terminal. WebGL is executed by the GPU of a mobile device. Codes executable in the GPU need to be prepared, and these codes are provided in a function pair. Each of the pair is called Vertex Shader and Fragment Shader. They are designed in GLSL (Graphics Library Shader Language) which is a language as strict as C/C++. These two as a set are called Shader Program. Vertex Shader calculates Vertex locations. Depending on the output location, WebGL can rasterize various types of primitives such as dot, line, and triangle. As these primitives are rasterized, Fragment Shader function is called on in the second step. Fragment Shader calculates colors of every pixel of the current primitives. Every data set that functions should access need to be provided to the GPU. Shader may receive data in any of the following 4 ways: Attribute and Buffer (location, normal line, texture coordinate , peak color), Uniform (global variable), Texture (image data), and Varying (rendering). Finally, Framebuffer converts memory bit maps into video signals that are displayed on the terminal monitor [6-8].

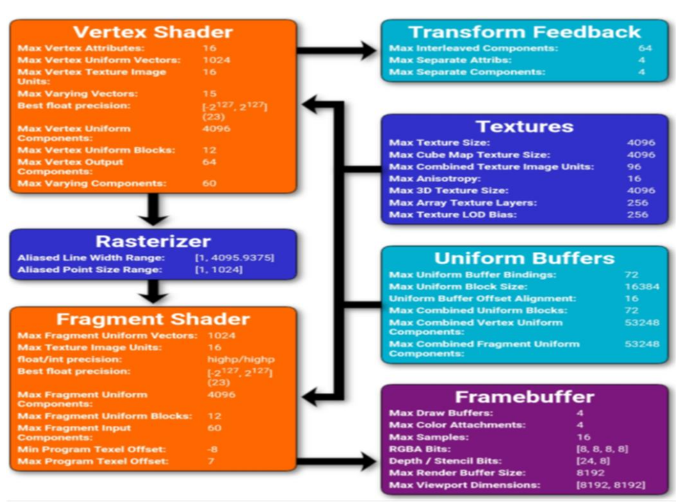


Figure 3. WebGL Report (webGL 2.0, Android 9, SM-G977N)

3. EXPERIMENTAL ENVIRONMENTS

For experiments in the same environments with previous studies, files were generated in each format of STL, OBJ, FBX, and glTF by means of the Blender. quantitative data was extracted depending on the loading time of each 3D file format in a mobile web. OBJ was loaded onto the mobile web browser by means of OBJ_Loader.js and MTL_Loader.js. STL was loaded to the mobile web browser by means of STL_Loader.js. For FBX, FBX_Loader.js was utilized. glTF could be loaded to the mobile web by means of glTF_Loader.js. In order to secure objective standards for comparison, the step of uploading through the Loader was omitted. A website was designed in a hosting server (kwonlab.or.kr), and then 3D objects shown in Figure 4 were uploaded to it and downloaded to a mobile device.

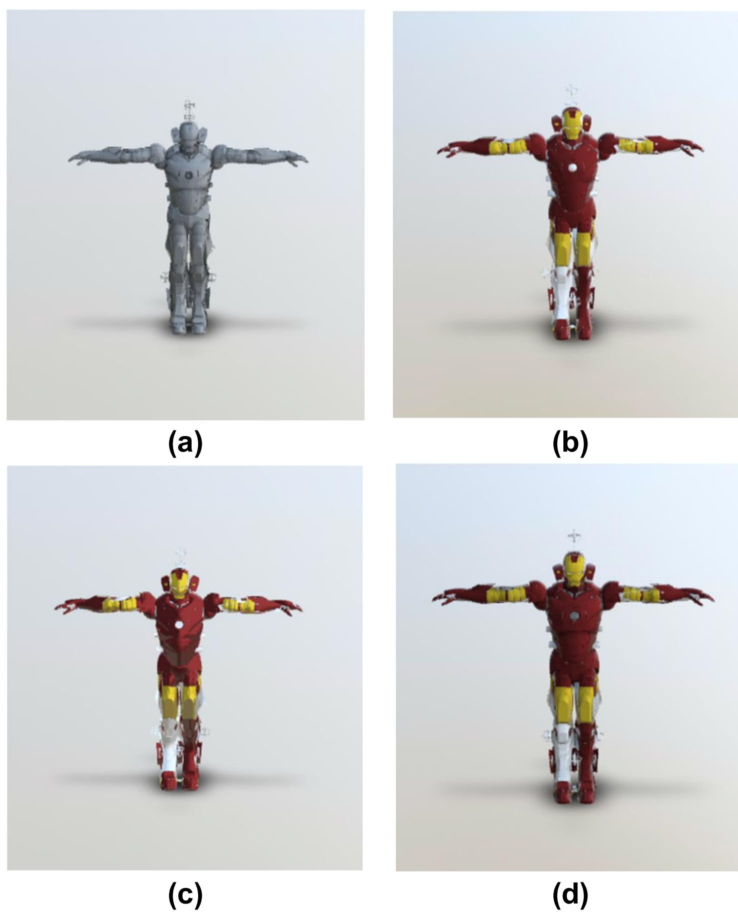


Figure 4. 3D Model Example

(a) STL (Verts: 104,963, Faces: 210,424, Tris: 210,424)
(b) OBJ (Verts: 129,757, Faces: 149,827, Tris: 217,038)
(c) FBX (Verts: 129,757, Faces: 149,827, Tris: 217,038)
(d) glTF (Verts: 131,044, Faces: 217,050, Tris: 217,050)

Figure 5 illustrates the 4G and 5G system utilized to measure the time of loading 3D model data on the mobile web by means of a development tool available in Google Chrome and Microsoft Edge. The development tool shows the time of loading to the mobile web comparatively. Major panels often used for debugging include Elements, Console, Network, and Sources. In experiments, the Network panel was utilized to measure the loading time of 3D model data.[9-10]

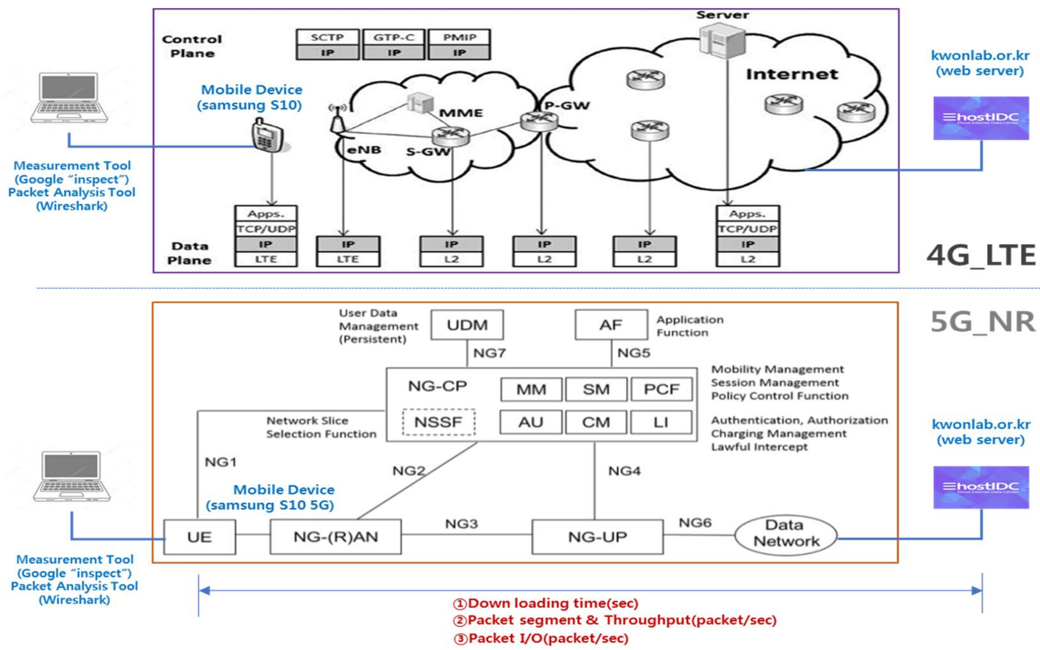


Figure 5. 4G vs 5G Experimental System Configuration

Table 1 shows the browser development tool, experiment site, mobile WebBrowser/WebGL, hardware performance, OS, and packet measuring program that were used in experiments.[11]

Table 1. Experiment environment

Class ification	Information	
	4G_LTE	5G_NR
Mobile Hardware	SM-G975N(samsung gallaxy S10)	SM-G977N(samsung gallaxy S10 5G)
Measurment Tool	Google Development Tool "inspect"	Google Development Tool "inspect"
	Taosoftware "tpacketcapture"	Taosoftware "tpacketcapture"
Experimental Site	https://kwonlab.or.kr	https://kwonlab.or.kr
Used Browsers	Google "Chrome", Microsoft "Eege"	Google "Chrome", Microsoft "Eege"
Mobile WebGL	WebGL 1.0 (OpenGL ES 2.0)	WebGL 1.0 (OpenGL ES 2.0)
Mobile OS	Android 9	Android 9
Analysis Program	Dump cap(Wireshark) 2.6.4	Dump cap(Wireshark) 2.6.4

In order to configure the same environment with a previous study [11], every experimental tool except mobile devices was the same 4G or 5G devices. The wireless internet communication conditions between the mobile device and web server were measured 10 times in the same place. Table 2 shows the average of the 3 different quality values of wireless internet access measured at mobile base stations. The value of the download throughput in the case of the 5G NR was 1.071Gbps; that of the upload throughput was 68.6 Mbps; and that of Ping (Latency) was 23.1 msec. As to the wireless internet access quality of the 4G LTE network in the previous experiment, that of the download throughput was 245Mbps; that of the upload throughput was 32.2 Mbps; and that of Ping (Latency) was 29.9 msec.

Table 2. Wireless Internet Network Quality at Experiment Points

No.	DownLoad (Mbps)		UpLoad (Mbps)		Ping(Latency, msec)	
	4G_LTE	5G_NR	4G_LTE	5G_NR	4G_LTE	5G_NR
Average	**245**	**1071**	**32.2**	**68.6**	**29.9**	**23.1**
1	227	1080	43.4	71.8	36.2	23.0
2	244	1049	40.9	71.8	30.8	23.1
3	244	1040	38.6	68.2	30.4	22.8
4	238	1080	27.0	69.4	31.2	23.2
5	242	1094	26.8	71.3	29.0	23.4
6	254	1064	26.1	69.7	28.4	22.9
7	231	1075	24.7	65.5	28.3	23.0
8	225	1076	25.1	71.5	29.2	23.0
9	291	1097	25.8	56.2	27.9	23.2
10	251	1050	43.4	70.3	28.0	23.1

4. Experiment and Result

4.1. Experiment

3D model data loading was performed by means of Google Chrome and Microsoft Edge in a Samsung galaxy S10 5G (SM-G977N) terminal. 3D model data was converted into each of the 3D file formats: STL, OBJ, FBX, and glTF. The loading time was measured based on the three standards: DOMcontentloaded, load, and Finished [12]. The DOMcontentloaded event indicates the timing when the DOM tree was completed but an external resource (img etc..) had yet to be loaded. The load event indicates the time when every resource (img, style, script etc..) was loaded onto the browser. The Finish event indicates that the time until 3D model data loading started and ended after every source was downloaded.

Table 3 shows in a table the loading time in Google Chrome and MS Edge. In comparison with STL and OBJ, the 3D model data loading time of FBX and glTF was relatively short in both of the browsers (Chrome, Edge). In the Edge browser, the loading time of the STL and OBJ models was 9.36 sec. and 14.29 sec. longer respectively. After 8 repetitions of the test, the standard deviation was 381 msec., which indicates that the value of FBX was the lowest and thus its loading characteristics were stable in general.

Table 3. Browser-specific loading time (msec)

Browser		Domcontent Loaded	Load	Finished	Standard Daviation
Chrome	**STL**	831	829	3,665	643
	OBJ	792	791	4,558	811
	FBX	778	776	4,786	**381**
	glTF	658	657	4,303	729
Edge	STL	783	782	9,360	423
	OBJ	835	834	14,299	**214**
	FBX	830	828	5,214	841
	glTF	741	739	**4,801**	995

Figure 6 shows in a graph the loading time in Google Chrome and MS Edge. As indicated by this data, the loading time of OBJ and STL in the case of Edge was longer than that of FBX, and glTF in general.

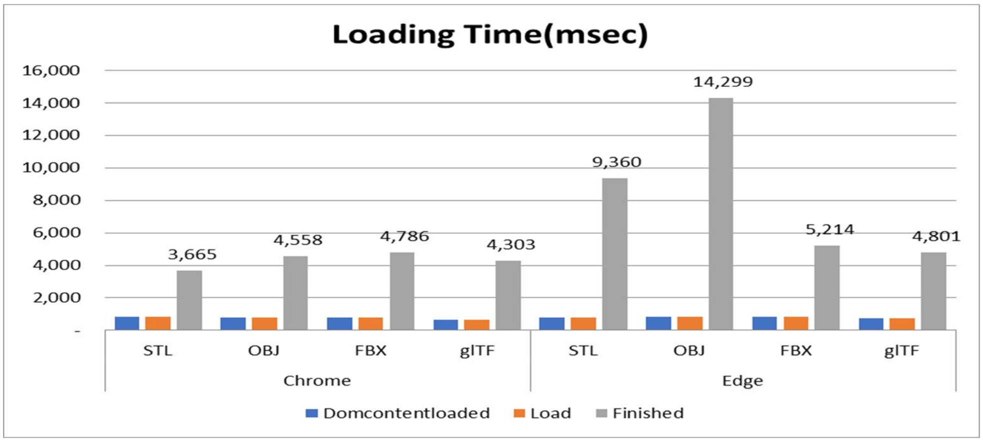


Figure 6. Browser-specific loading time in graph

Figure 7 shows the packet segment length of 3D model data and the average bit per second. The packet segment length indicates the size of data loadable onto the TCP except the IP header and TCP header. The brown line indicates the average bit per second.

Figure 7 (a) shows the packet segment length of 3D model data and the average bit per second of STL in Chrome. On the brown line, the packet transmission involved a delay between 0.4 and 0.6 sec. Figure 7 (b) shows the packet segment length of 3D model data and the average bit per second of STL in Edge. On the brown line, the packet transmission involved a delay at around 0.5 sec. Figure 7 (c) shows the packet segment length of 3D model data and the average bit per second of OBJ in Chrome. On the brown line, the packet transmission involved a delay between 0.6 and 0.9 sec. Figure 7 (d) shows the packet segment length of 3D model data and the average bit per second of OBJ in Edge. On the brown line, the packet transmission involved a delay between 0.5 and 1 sec. Figure 7 (c) shows the packet segment length of 3D model data and the average bit per second of FBX in Chrome. On the brown line, the packet transmission involved a delay between 0.6 and 0.9 sec. Figure 7 (f) shows the packet segment length of 3D model data and the average bit per second of FBX in Edge. On the brown line, the packet transmission involved a delay between 0.4 and 0.6 sec. Figure 7 (g) shows the packet segment length of 3D model data and the average bit per second of glTF in Chrome. On the brown line, the packet transmission involved a delay between 0.4 and 0.7 sec. Figure 7 (h) shows the packet segment length of 3D model data and the average bit per second of glTF in Edge. On the brown line, the packet transmission involved a delay between 0.2 and 0.5 sec. Figure 6 shows that the loading time is varied depending on the network quality. As the throughput is variable in the Chrome browser, the loading time increases consistently particularly in the case of STL/OBJ files whose 3D model size is relatively large. The brown line between 0.2 and 0.9 sec. in Figure 6 indicates a delay in packet transmission although more investigation is required to clarify why the packet transmission is delayed depending on 3D file formats.

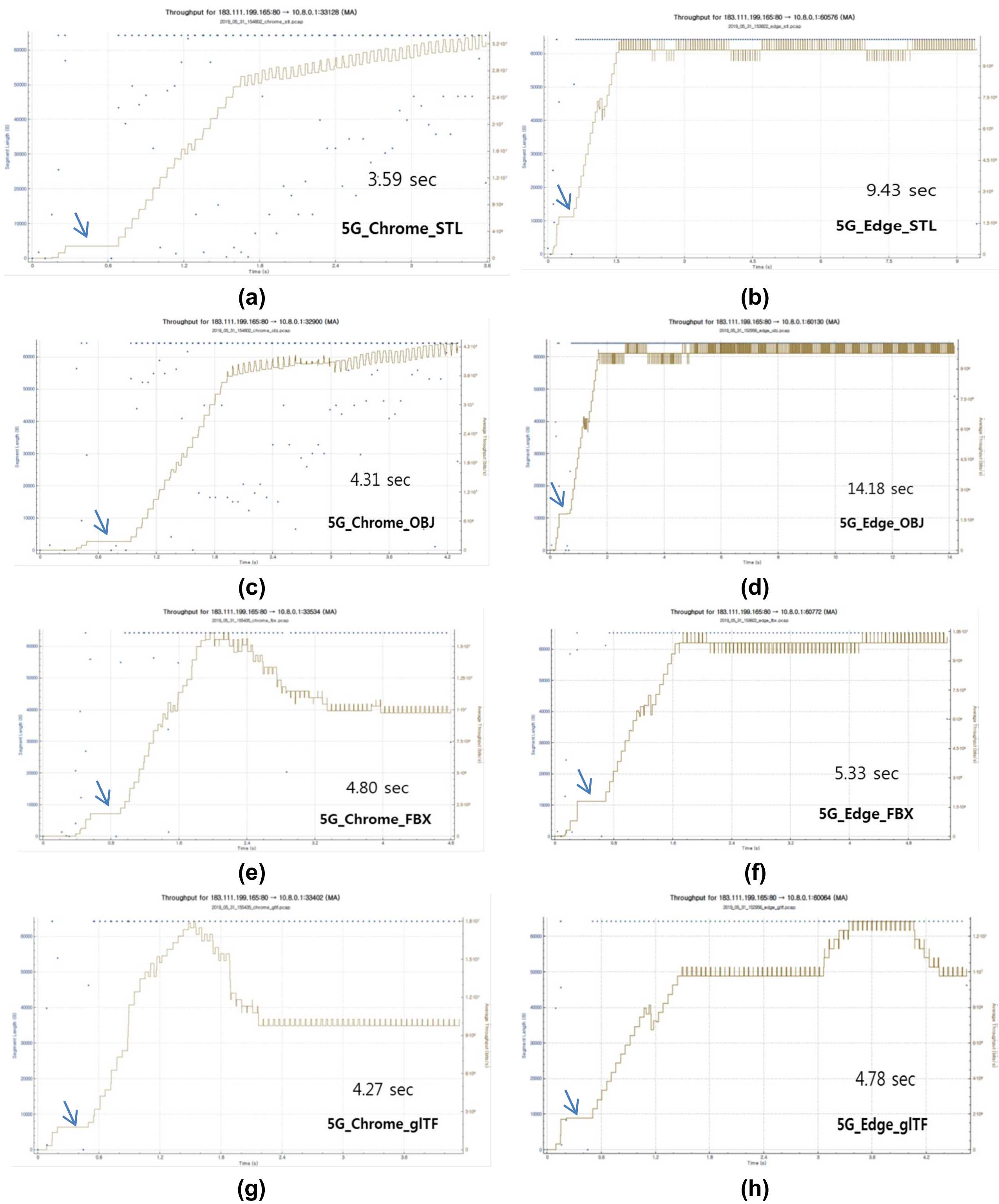


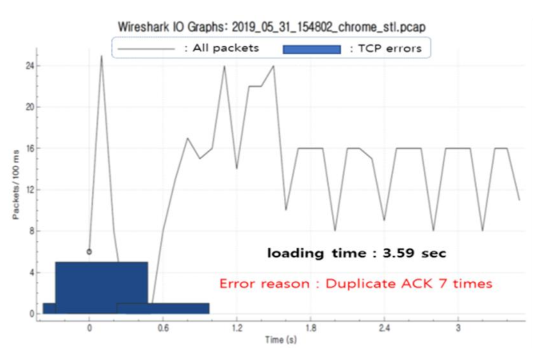
Figure 7. Packet Segment and Throughput Graph

(a) Chrome_STL, (b) Edge_STL, (c) Chrome_OBJ, (d) Edge_OBJ,

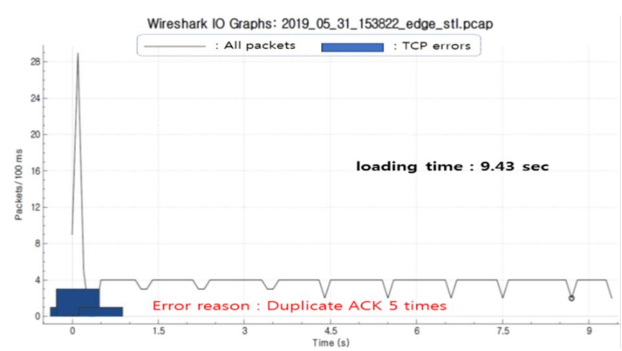
(e) Chrome_FBX, (f) Edge_FBX, (g) Chrome_glTF, (h) Edge_glTF

Figure 8 shows the beginning and end of packet transmission based on the quantity of packets transmitted per sec. in order to clarify the cause of packet transmission of 3D objects. Figure 8 (a) shows that 7 TCP errors occurred from the beginning of STL data packet transmission in Chrome up to the point of 0.6 sec.

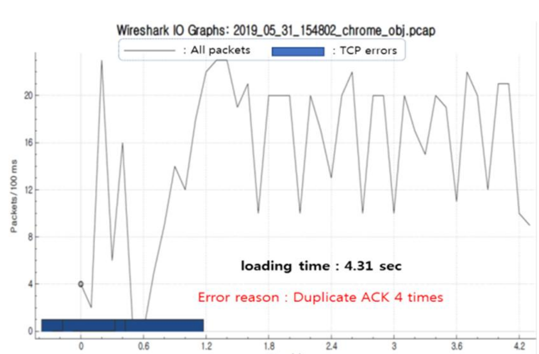
The process was completed in 3.59 sec. Figure 8 (b) shows that 5 TCP errors occurred from the beginning of STL data packet transmission in Edge up to the point of 0.5 sec. The process was completed in 9.43 sec. Figure 8 (c) shows that 4 TCP errors occurred from the beginning of OBJ data packet transmission in Chrome up to the point of 0.9 sec. The process was completed in 4.31 sec. Figure 8 (d) shows that 9 TCP errors occurred from the beginning of OBJ data packet transmission in Edge up to the point of 0.9 sec. The process was completed in 14.18 sec. Figure 8 (e) shows that 4 TCP errors occurred from the beginning of FBX data packet transmission in Chrome up to the point of 0.8 sec. The process was completed in 4.8 sec. Figure 8(f) shows that 3 TCP errors occurred from the beginning of FBX data packet transmission in Edge up to the point of 0.8 sec. The process was completed in 5.33 sec. Figure 8 (g) shows that 5 TCP errors occurred from the beginning of glTF data packet transmission in Chrome up to the point of 0.6 sec. The process was completed in 4.27 sec. Figure 8 (h) shows that 1 TCP error occurred from the beginning of glTF data packet transmission in Edge up to the point of 0.5 sec. The process was completed in 4.78 sec. It turned out that there were 2 major causes of TCP errors: The first was duplicate ACK involving packet loss causing retransmission. The second cause was streaming through various paths as the system was out of order, causing a transmission delay while the receiver waits. The third cause was a connection reset that resulted from password resetting upon the completion of HTTP 3-handshake.



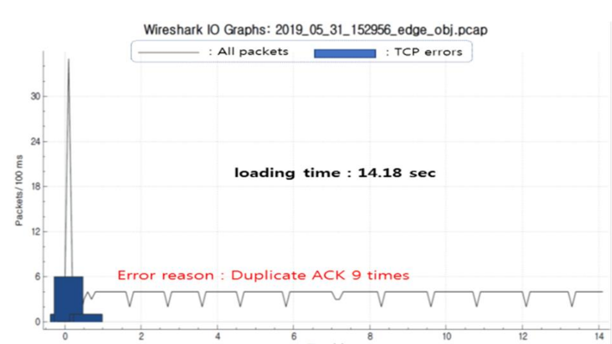
(a)



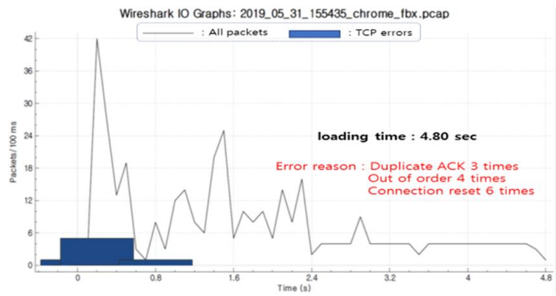
(b)



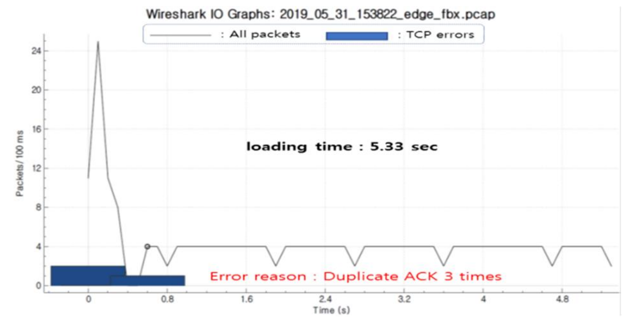
(c)



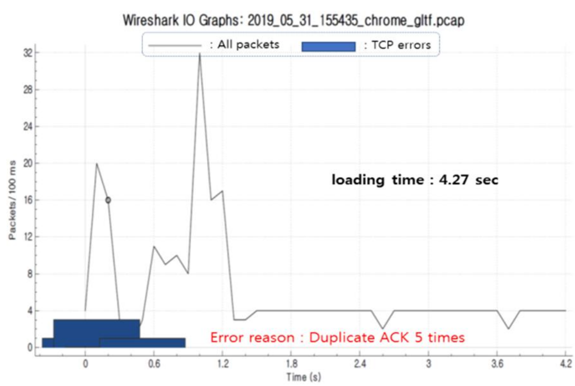
(d)



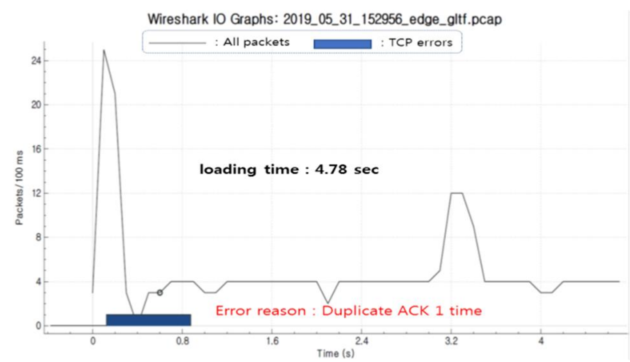
(e)



(f)



(g)



(h)

Figure 8. Packet I/O graph

(a) Chrome_STL, (b) Edge_STL, (c) Chrome_OBJ, (d) Edge_OBJ,

(e) Chrome_FBX, (f) Edge_FBX, (g) Chrome_glTF, (h) Edge_glTF

The result of this experiment (5G NR) was compared with that of the previous study (4G LTE) as in Figure 9. As this table shows, the loading time was different depending on the 3D model file formats. In this experiment (5G NR), the loading time was longest in the Edge browser and in the OBJ file format. In the case of 4G LTE, STL and OBJ file formats showed the longest loading time in both Chrome and Edge browsers [7]. As the number of transmitted packets per 100 msec was compared in the packet I/O graph of Figure 7, that of the Chrome browser was variable depending on the traffic volume while that in the Edge browser was fixed to 4 packets/100 msec. As the model size was large, the loading time was affected to the corresponding degree of significance. Hence, the Edge browser showed a constant loading time over different model sizes (OBJ:16MB > STL:10.6MB > FBX:5.8MB > glTF:5.5MB) regardless of the changing experimental environments of the 4G/5G mobile communication network.

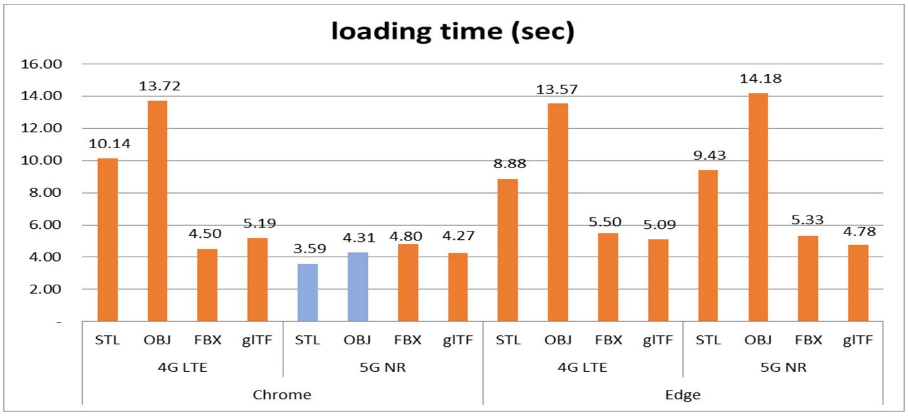


Figure 9. Performance Comparison Result

4.2 Result

This paper compares characteristics of 4G and 5G relecommunication networks, explores the structure of Mobile WebGL, and compares the mobile loading time among 3D file formats such as STL, OBJ, FBX, and glTF. Experimental results may be summarized as below: First, the packet transmission procedures in mobile WebGL were varied depending on the browsers. In Figure 7 and Figure 8, the packet transmission rate of the Chrome browser was variable depending on the size of the 3D model while that of the Edge browser was fixed to 4 packets/100 msec. As the model size was large, the loading time was affected to the corresponding degree of significance. Second, the glTF file format demonstrated the best characteristics as shown in Table 3 and Figure 9. glTF reduced overhead by using JSON and binary files, starting parsing earlier. For large-size data such as geometry, the format efficiency was maximized by storing it in a binary file. This is the reason why Open GL, Facebook, Google, and Microsoft started to support glTF format for 3D model data since glTF was released, and the base has been expanded accordingly. Third, in the case of mobile packet transmission, retransmission due to packet loss in wireless sections occurred more frequently than in wired transmission. As shown in the packet I/O analysis in Figure 7, there was no difference in packet retransmission frequency between 4G LTE and 5G NR. As the 3D model file size increases, the general loading time is affected accordingly. In the case of Chrome, however, the model size (16 MB) of OBJ files was the largest but the loading time (4.31 sec., 4.27 sec.) was similar to that of glTF files (5.5 MB). This indicates that the 3D model loading in the wireless section was significantly improved in mobile WebGL of 1Gbps (5G NR).

5. Conclusion

glTF was demonstrated as a 3D modeling file format suitable for representation of various realistic contents such as augmented reality contents in mobile settings. As to transmission errors in a wireless section of the mobile communication network, a similar level of packet retransmission occurred in both 4G and 5G systems, and the occurrence rate was higher than that of wired sections. In the case of Chrome in the present 5G NR commercial network, the loading time is shortened stably regardless of the file size. As the latency time of the 5G mobile communication network is shortened down to 10 msec. in the future, it will be possible to analyze Duplicate ACK more thoroughly, which was regarded as another cause of packet transmission delay and 3D model packet loading time extension. It is expected that the findings of this study can be referred to as quantitative data not only by users who experience inconvenience due to the long loading time of 3D models but also by researchers examining obstacles to Mobile WebGL continually.

ACKNOWLEDGEMENT

The present Research has been conducted by the Reasearch Grant of Kwangwoon University in 2019.

REFERENCES

[1] Z. Zhao, K. He, and R. Du, "The Simulation of Scara Robot Based on OpenGL and STL," 2011 Second International Conference on Mechanic Automation and Control Engineering (SICMACE), pp. 5429-5432, Jul. 15-17, 2011. DOI: 10.1109/MACE.2011.5988251.

[2] Jahanzeb Hafeez, Seunghyun Lee, Soonchul Kwon, and Alaric Hamacher, "Image Based 3D Reconstruction of Texture-less Objects for VR Contents," International Journal of Advanced Smart Convergence (IJASC), Vol. 6, No. 1, pp. 9-17, Jun 2017. DOI: 10.7236/IJASC.2017.6.1.9.

[3] Jongdeug Kim and Taehyun Jeon, "An Effective Solution for the Multimedia Telephony Services in Evolving Networks," International Journal of Advanced Smart Convergence (IJASC), Vol. 2, No. 1, pp. 24-26, Feb2013. DOI: 10.7236/IJASC.2013.2.1.024.

[4] SKT to offer 500Mbps – Fastest speed by LTE so far, *https://www.netmanias.com/en.*

[5] Recommendation ITU-R M.2083-0, *https://www.itu.int/rec/R-REC-M.2083-0-201509-I/en*

[6] WebGL, *https://webglfundamentals.org/webgl/lessons/webgl-fundamentals.html*.

[7] M. Zhao and J. Zhang, "Rapidly Product and Optimize Facial Animation Methods for 3D Game," International Conference on Internet Computing in Science and Engineering (ICICSE), pp. 136-139, Jan. 28-29, 2008. DOI: 10.1109/ICICSE.2008.15.

[8] Geonhee Lee, Pyeong-ho Choi, Hwa-seop Han, Seunghyun Lee, and Soonchul Kwon, "A Study on the Performance Comparison of 3D File Formats on the Web," International Journal of Advanced Smart Convergence (IJASC), Vol. 8, No. 1, pp. 65-74, Mar 2019. DOI: doi.org/10.7236/IJASC.2019.8.1.65.

[9] Geonhee Lee, Seunghyun Lee, and Soonchul Kwon, "A study on Compression of 3D Model Data and Optimization of Website," Journal of Engineering and Applied Sciences (JEAS), Vol. 14, No. 1, pp. 3934-3937, 2019. DOI: 10.3923/jeasci.2019.3934.3937.

[10] Adapted from TR23.799 v0.8.0, *https://www.3gpp.org/DynaReport/23-series.htm*

[11] Duckkyoun Nam, Daehyeon Lee, Seunghyun Lee, and Soonchul Kwon, "Performance Compression of 3D File Formats on a Mobile Web Brower," International Journal of Broadcasting and Communication (IJIBC), Vol. 11, No. 2, pp. 31-42, 2019. DOI:10.7236/IJIBC.2019.11.2.31

[12] P. H. Shroff and S. R. Chaudhary, "Critical Rendering Path Optimizations to Reduce the Web Page Loading Time," International Conference for Convergence in Technology (ICCT), pp. 937-940, Apr. 7-9, 2017. DOI: 10.1109/I2CT.2017.8226266.