

Green Ajax Implementation on the Wireless Local Area Network using Randomized Cue Applications

Ridwan Sanjaya, *Member, IACSIT*

Abstract—Green Ajax as a new approach in the web development has an aim to create interactive communication between server and client but avoiding the useless redundant requests. Some researches have been conducted to prove the benefit of Green Ajax, compared to Classic Ajax. The results of experiment show that the performance of Green Ajax is better than Classic Ajax in almost of all conditions. These results strengthen the conclusion that Green Ajax is a good approach in the web-based application development.

However, some further researches have to be conducted in different environments to prove that the conclusion is also valid for a wider variety of conditions. One of those conditions is the wireless Local Area Network which is now very commonly available in schools, hotels, offices and business centers. Thus, the implementation of Green Ajax for the applications that support business, not just an experiment, can be realized immediately by using Randomized Cue Applications.

This paper will focus on the analysis of Green Ajax performance on the wireless Local Area Network. The program discussed will demonstrate the ability of Green Ajax in providing a good interaction between users and the web server. The results of the experiment confirm that the Green Ajax is suitable for the applications on the wireless Local Area Network. Network traffic reduction is obtained as the final result in the use of Green Ajax.

Index Terms—Green Ajax, network traffic, web-based application, wireless LAN.

I. INTRODUCTION

Green Ajax is an enhancement of Classic Ajax which has been recognized in the web application. Inheriting from the origin, Green Ajax also has the ability to decrease the bandwidth consumption by delivering data only in the specific part of web page [8]. The user interfaces will display any information without any interruption on screen because the communication between web application and server is done in the background.

The improvement is located in the ability of web application to receive notification from the server whenever an update or a change of data occurs. When the application receives the server's cue, the application will then send out a request [7]. The applications will display the latest

information without any involvement from the user. This method reduces required bandwidth below than necessary for data updating.

In the previous experiments done in the wired Local Area Network, Green Ajax has been proven as a suitable approach for web-based application in variety conditions. When the server updates the data in unpredictable time, Green Ajax applications are always able to be a better approach in displaying the latest data. Green Ajax sends fewer requests to the server compared to Classic Ajax. The less bandwidth consumption of Green Ajax reduces the network traffic [7].

Furthermore, Green Ajax is still able to display the latest data using smaller amount of requests than Classic Ajax even though the server updates the data at the scheduled time and Classic Ajax uses the same interval time of data updating in the server for its TTR [8]. In addition, Green Ajax is also able to save the bandwidth when the frequency of updates is very frequent (High Frequency Update). The same benefit can be seen when the frequency of update activity is medium (Medium Frequency Update).

Even though it is insignificant, Green Ajax uses larger bandwidth than Classic Ajax when the frequency of data updating is rare (Low Frequency). But in overall results, Green Ajax is dominating the success in comparison with Classic Ajax.

In the experiments using Fuzzy-based applications, the server extends the range of time for data updating on each classification of frequency [9]. The extension of range is done because there is no exact crisp between each classification in the reality. The Fuzzy exists between the High to Medium Frequency Update and Medium to Low Frequency Update. With the expansion area, this experiment became more real than previous experiments. As the results, the margin of bandwidth consumption between Green Ajax and Classic Ajax is very significant. Green Ajax uses less bandwidth than Classic Ajax when the frequency of updates is often and medium. When the frequency of updates is rare, Green Ajax uses slightly bigger bandwidth but insignificant compared to the Classic Ajax.

In this research, the server will perform the update in unpredictable time but limited 5 minutes as the maximum interval time of update. One client who uses Classic Ajax will perform data requests to the server periodically. The client uses a shorter time than the maximum value of data updating interval time on the server, which is 2 minutes. At the same time, the four clients who use the Green Ajax depend on the cue sent by the server. If the cue is received by the client, the

Manuscript submitted on March 18, 2011.

Ridwan Sanjaya is with the Graduate School of Information Technology, Assumption University, Bangkok, Thailand as the Ph.D. candidate, on leave from the Computer Science Faculty, Soegijapranata Catholic University, Semarang, Central Java Province, Indonesia (e-mail: ridwan.sanjaya@gmail.com)

client will make a request to the server. Even though the experiments use only five clients as seen on Fig. 1, it is possible to add more clients if the organization has more than five clients.

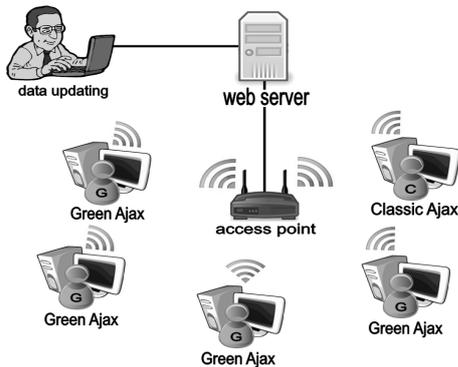


Figure 1. Five clients of Green Ajax and 1 client of Classic Ajax

The experiment was performed on a wireless Local Area Network using the assumption: the clients are located near the access point and there is no network disruption during the experiment.

The structure of the discussion in this paper is divided into five sections. Introduction and Literature Review will be discussed in section 1 and section 2. Then, Green Ajax assessments in wireless LAN environments will be discussed in section 3. The results of experiment on each client will be analyzed in section 4 and concluded in section 5.

II. LITERATURE REVIEW

The discussion of Green Ajax and Classic Ajax is not far from the discussion of client-side scripting. Both of them are built using JavaScript as a popular client-side language of scripting in the web programming. JavaScript is the client side scripting, which can control the interaction between user and the web page by calling the XMLHttpRequest to request data to the server without disturbing the web page [4]. After the data is obtained from XMLHttpRequest, the JavaScript will be used to manipulate the contents of a web page in Document Object Model (DOM) [1].

XMLHttpRequest is the main technology that makes Ajax engine able to request to the server and receive data from the server [10]. DOM is a structure within a web page, in which its content and visibility can be modified by using JavaScript [5]. When the value of the object is changed, the specific parts of the web page are also changed. The benefit, web

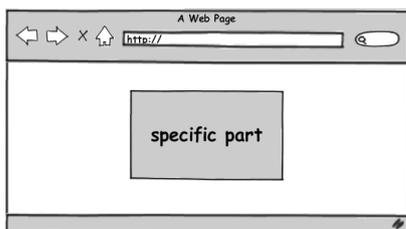


Figure 2. Delivering data only in the specific part of web page

application does not need to replace the entire page when requesting the latest data from the server. Only the needed data will be delivered to the specific part of the web page as

seen on Fig. 2. It will reduce the bandwidth and increase the speed to display the content of web page.

However, the server will not send the data to the client automatically. JavaScript must issue requests from the client to the server by calling the methods in XMLHttpRequest then wait responses from the web server. It is common in practice that web application will repeat the requests at regular user-definable intervals known as Time to Refresh (TTR) [8] as seen on Fig. 3. If the responses are the same as the old data, the same information will be displayed on the client side. These unnecessary requests will waste the request time, response time and resources. In some circumstances, the interactivity of web application has to be paid with wasted bandwidth.

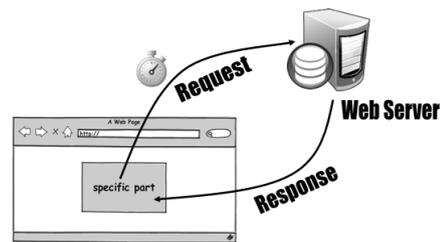


Figure 3. The content is requested from the server periodically

Green Ajax solves the problem by providing a new approach to send a cue from web server to the clients whenever an update arises. After the client receives a cue from web server, the client will issue the request to the web server [7]. Fig. 4 shows how the Green Ajax works that has been explained above.

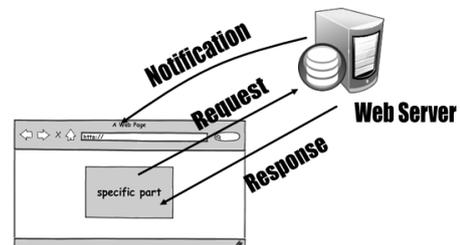


Figure 4. The content is requested from the server based on notification

This method reduces required bandwidth below than necessary for data updating. It is a suitable approach for bandwidth saving in any conditions, including infrequent and frequent update applications. From the experiment, Green Ajax is proved to cut successfully required bandwidth in presenting the latest updates from the server.

Fig. 5 shows when the administrator updates the database on the server, a cue will be released to the client by the server. Then, the client will send a request to the server and a new content will be received by the client from the server. The latest contents will appear inside a web page that is displayed on the screen the client.

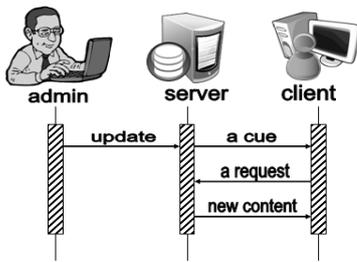


Figure 5. The cue arises when the administrator update the data

From the results, Green Ajax shows the best performance in Infrequent Update Applications. Those experiments were tested on the case of random and unpredictable update time on the web server. Green Ajax also shows its best performance in the case of fixed update time on the web server. Majority, Green Ajax can be used in any situation of update time.

TABLE I. EXPERIMENTS RESULTS OF THE CASE OF FIXED UPDATE TIME ON THE WEB SERVER

#	Situation	Better Results		
		Bandwidth	SRP	DLP
1	High Frequency Update	Green	Green	Both
2	Medium Frequency Update	Both	Green	Both
3	Low Frequency Update	Classic	Both	Both

In the experiments using Fuzzy-based Applications, Green Ajax is still able to be the most suitable approach in the real situation [9]. Those experiments have five time-zone-update times which consist of three independent zones and two strips.

TABLE II. EXPERIMENT RESULTS OF FUZZY-BASED APPLICATION

#	Scenario	
	Situation	Better Approach
1	High Frequency Update	Green
2	Medium Frequency Update	Majority Green
3	Low Frequency Update	Classic

The comparison of performance between Green Ajax and Classic Ajax will be done on the wireless Local Area Network. The client will connect to the server via an access point. Wireless Local Area Network is using high frequency radio waves 2.4 GHz (802.11b, 802.11g) or 5 GHz (802.11a) for its communications [11]. Wireless Local Area Network became popular because every computer is now equipped with Wi-Fi facility. The clients will join easily the Local Area Network without having to plug or unplug the network. Once the client is identified and validated on the network, the client can be connected to other computers in the network.

III. IMPLEMENTATION

In the Green Ajax implementation, a routine program is required in the server side to trigger a cue transmission to the client whenever an administrator does an updating. The routine program can be integrated to the update program, or can be separated modules which can be used together by

other programs. A one byte character will be used as a cue that is sent to the clients. The reason of using one byte character is to limit the bandwidth usage, keep the client secure, and avoid harmful applications.

Each client on the wireless Local Area Network has their own identity in the form of an IP address. When the client connects to the server, the IP address of the client will be recorded by the server. That identity will be used as a target to receive the cue from the server when an administrator does an update. The illustration of the implementation can be seen in Fig. 6.

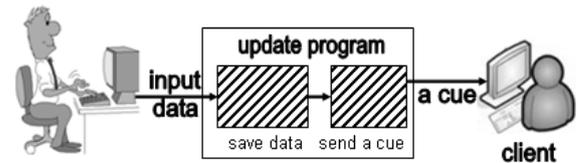


Figure 6. A routine program to trigger a cue transmission

The following PHP codes describe how the update program works. The RandomRefresh() function will set the next update time, CreateData() function will create random data for the experiments, and SendCue() function will trigger a cue transmission to the clients.

```
<?php
include "GreenAjax.php";
$myRandomApp = new GreenAjax();
$myRandomApp->RandomRefresh();
$myRandomApp->CreateData();
$myRandomApp->SendCue();
?>
```

The small program on the client has the duty to receive the cue. It can be either a plug-in or extension that is installed in the web browser. POW (Plain Old Webserver) is an example of plug-in used in this research. A small program on the client will interpret the character received as a command to make a request to the server as seen in Fig. 7. This method is more effective than using a repetition of the request based on TTR. Furthermore, new data will be sent to the client after the server receives such request.



Figure 7. A small program embedded on the web browser

To make the plug-in works, the application has to be connected with the plug-in. The following PHP codes describe how the main program works. The initAjax() function will enable Ajax on the application. Then, initPlugin() function will create a connection to the plug-in. The Listen() function will receive a cue to trigger the client in making request to the server.

```
<?php
include "GreenAjax.php";
$myClientApp = new GreenAjax();
```

```
$myClientApp->initAjax();
$myClientApp->initPlugin();
$myClientApp->Listen();
?>
```

The `initPlugin()` function above will listen the cue sent by the server. Whenever a cue is received, the client will request using the following code to update the data. The `readData()` function will read the new data on the server and the `Output()` function will print the result. The request will be done in the background by using `XMLHttpRequest`, `DOM`, and `JavaScript`.

```
<?php
include "GreenAjax.php";
$myDataApp = new GreenAjax();
$myDataApp->readData();
$myDataApp->Output();
?>
```

After sending a cue, the clients who are still requesting to the server will be recorded by the server using their IP address. This mechanism makes the server work effectively in sending the cue to the clients on the next update. A Session Time can also be used to monitor the clients who are accessing the server.

In this research, a Randomized Cue Applications are used as a simulator to produce an update in the random time. Time decision of the next update is based on the random value generated by the application. The results obtained can describe the actual conditions when Green Ajax is used. Then, the bandwidth saving will be calculated using Firebug 1.5.4 as one of additional plug-ins on Mozilla Firefox 3.5.14. Firebug as seen in Fig. 8 has functions to debug, edit, and monitor any object on a web page [6]. In this research, its function to monitor is used to calculate the size of data transfer from/to the server.

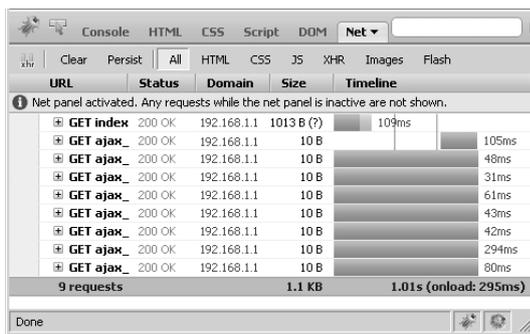


Figure 8. Mozilla Firefox 3.5.14 and Firebug 1.5.4

IV. RESULTS AND ANALYSIS

The bandwidth consumption of Randomized Cue Applications will be compared with the Classic Ajax Application's bandwidth consumption. The Classic Ajax Application uses 2 minutes TTR to request the update data from the server.

From the results, the number of times data that had been received by the client (Rcv), and the number of times requests that had been issued by the server (Req) were compared to get the Successful Receptive Percentage (SRP)

[7]. Data Loss Percentage (DLP) were calculated by comparing the number of times data was not received timely by the client (Los) and the number of times update data was activated by the server (Upd).

TABLE III. SRP AND DLP

Results ^a	Green Ajax				Classic Ajax
	1	2	3	4	
SRP (%)	100.00				69.54
DLP (%)	0.00				18.60

a. Experiment time: 5 hours

Table III shows the SRP of Green Ajax applications are 100%. It shows no wasting requests had been issued by Green Ajax. However, Classic Ajax only gets 69.54% data. It shows there are 30.56% requests to the server but do not get new data (wasting requests). In another result, DLP of Green Ajax applications are 0%. It means all data has been received by each Green Ajax application. However, the Classic Ajax gets 18.60% DLP. It shows there are 18.60% data updating from server that un-received by Classic Ajax.

The visualization results of experiments to compare Green Ajax and Classic Ajax can be seen on the Fig. 9. The SRP bars show that all SRP of the Green Ajax applications are very high compared to SRP of the Classic Ajax application which is only 69.54%. The DLP bars indicate that the DLP from all Green Ajax is very low compared to the Classic Ajax which can reach 18.60%.

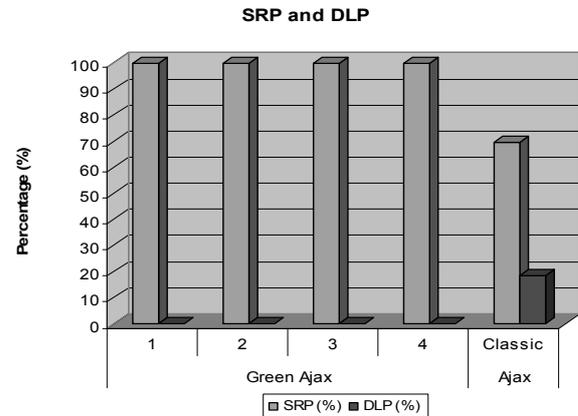


Figure 9. The result of SRP and DLP

The bandwidth consumption of five hours experiment can be seen on Table IV. It shows that bandwidth consumption of Green Ajax applications is smaller than Classic Ajax. In the Data Loss, there are 14,400 bytes data transferred by server which are not received by the clients. It is different from Green Ajax that does not lose any data.

TABLE IV. BANDWIDTH CONSUMPTION AND DATA LOSS (IN BYTES)

Results ^a	Green Ajax				Classic Ajax
	1	2	3	4	
Bandwidth	77,529				90,729
Data Loss	0.00				14,400

a. Experiment time: 5 hours

Based on table IV, the visualization of bandwidth consumption and data loss of each application can be seen on Fig. 10. The bars of bandwidth consumption of Green Ajax are shorter than the bar of bandwidth consumption of Classic

Ajax. Other bars show that data loss of Green Ajax applications is very low compared to Classic Ajax which never receives 14,400 bytes data on the server.

The above results show the benefit of Green Ajax Implementation on the wireless Local Area Network has been successfully proved by using Randomized Cue Applications. Even though the cue has a size that can lead to ineffectiveness, the results indicate that the cue helps the application to request more effectively.

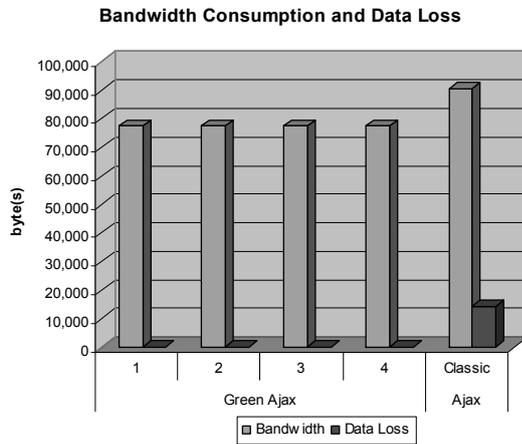


Figure 10. The Bandwidth Consumption and Data Loss

V. CONCLUSIONS

Green Ajax performs excellently on the wireless Local Area Network. The implementation of Green Ajax is not limited to the media used for data transmission. Green Ajax performance in wireless Local Area Network is as good as tested on the wired LAN. By using Randomized Cue Applications, the real-time data will be showed on screen using less bandwidth. The bandwidth reduction is caused by two main reasons. First, not all portion of page should be updated. The technique to update on specific part of page can be done by collaboration between the XMLHttpRequest, DOM, and JavaScript [2]. Second, the number of requests to the server has been reduced by the clients. The cue from a server arises to tell the client to make requests. The technique will consume bandwidth only if there are any updates.

A small application on the client will receive a maximum of one byte character to trigger the client to make a request to the server. The reason to limit only one byte character is to limit the bandwidth usage, keep the client secure, and avoid harmful applications. The small application can be either a plug-in or extension that is installed in the web browser. POW is an example of plug-in used in this research. However, this application has common functions that are not used fully by Green Ajax. It is necessary to make a small application specifically required for the purposes of Green Ajax.

In further research, the scope of computer network will get more attention not only on a wired and wireless Local Area Network but also on a larger network. The identification of clients should not only depend on the IP address listed on the server that is sometimes not associated with a real IP address on the clients. However, using the current HTTP Protocol should be prioritized because it has been used widely. As a

result, the benefits of Green Ajax can be implemented widely on the computer network.

ACKNOWLEDGMENT

This manuscript is fully granted from the Ministry of National Education of the Republic of Indonesia since 2008.

REFERENCES

- [1] A. Marchetto, P. Tonella and F. Ricca, "State-Based Testing of Ajax Web Applications", *Proceeding of the 2008 International Conference on Software Testing, Verification, and Validation (ICST)*, Lillehammer, pp. 121-130, April 2008. doi: <http://dx.doi.org/10.1109/ICST.2008.22>.
- [2] C.W. Smullen and S.A. Smullen, "AJAX Application Server Performance", *Proceeding of the IEEE SoutheastCon 2007*, Richmond, VA, March 2007, pp. 154-158. doi: <http://dx.doi.org/10.1109/SECON.2007.342873>.
- [3] L. Zhijie, W. Jiyi, Z. Qifei, and Z. Hong, "Research on Web Applications Using Ajax New Technologies", *Proceeding of the 2008 International Conference Multimedia and Information Technology (ICMIT)*, Washington, DC, USA, December 2008, pp. 139-142. doi: <http://doi.ieeecomputersociety.org/10.1109/MMIT.2008.107>
- [4] N. Hanakawa and N. Ikemiya, "A Web Browser for Ajax Approach with Asynchronous Communication Model", *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI-06)*, Hong Kong, China, pp. 808-814. doi: <http://dx.doi.org/10.1109/WI.2006.30>.
- [5] N. Matthijssen, A. Zaidman, M.A. Storeyy, I. Bully and A.v. Deursen, "Connecting Traces: Understanding Client-Server Interactions in Ajax Applications", *Proceeding of the 2010 IEEE 18th International Conference on Program Comprehension (ICPC)*, Braga, Minho, pp. 216-225, July 2010. doi: <http://dx.doi.org/10.1109/ICPC.2010.14>.
- [6] P. Hope and B. Walther, *Web Security Testing Cookbook*, Sebastopo:, CA: O'Reilly Media, 2009.
- [7] R. Sanjaya, "Web traffic reduction for infrequent update application using Green Ajax", *Proceeding of The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, Chengdu, China, pp. 170-176, April 2010. doi: <http://dx.doi.org/10.1109/ICIME.2010.5477773>
- [8] R. Sanjaya, "Trade-off analysis for web application using Green Ajax", *Proceeding of IEEE International Conference on Management of Innovation and Technology (ICMIT)*, Singapore, pp. 1050-1054, June 2010. doi: <http://dx.doi.org/10.1109/ICMIT.2010.5492884>
- [9] R. Sanjaya, "Mobile Traffic Evaluation for Fuzzy-Based Application Using Green Ajax," *Proceeding of the 2011 IEEE International Conference on Information and Education Technology (ICIET)*, to be published.
- [10] R. Sanjaya and C. Brahmawong, "Distance Examination using Ajax to Reduce Web Server Load and Student's Data Transfer", *International Journal of the Computer, the Internet and Management (IJCIM)*, Volume 15 SP3, Assumption University of Thailand, pp. 24.1-6, November 2007.
- [11] T. Soungalo, L. Renfa, and Z. Fanzi, "Evaluating and Improving Wireless Local Area Networks performance," *Proceeding of the 2010 3rd International Conference on Information Sciences and Interaction Sciences (ICIS)*, pp.367-372, June 2010. doi: <http://dx.doi.org/10.1109/ICIS.2010.5534802>



Ridwan Sanjaya has been working as a lecturer in Soegijapranata Catholic University, Semarang, Indonesia since 2002. This author became a member of IEEE, IACSIT, and IEICE. He received the Master of Science in Internet and E-Commerce Technology (MS.IEC) degree from Assumption University, Bangkok, Thailand in 2007. Currently he is pursuing his Doctoral degree in Computer Information Systems (CIS) at Graduate School of Information Technology, Assumption University, Bangkok, Thailand. He has been publishing more than 60 books related to computer area such as Web Development with JSP, Graphic Engineering using PHP, PDF Report Development with PHP 5.0, Cross-Platform Computer Network Administration, Creative Digital Marketing, Business-Driven Information System, etc.