IN THE NAME OF ALLAH,

THE MOST GRACIOUS,

THE MOST MERCIFUL

JKAU: Comp. IT. Sci., Vol. **2,** 82 Pages (2013 A.D. / 1435 A.H.) ISSN: 1658-6336
Legal Deposit 1435/1028

Journal of

KING ABDULAZIZ UNIVERSITY

Computing and Information Technology Sciences

Volume 2

2013 A.D. / 1435 A.H.

Scientific Publishing Centre
King Abdulaziz University
P.O. Box 80200, Jeddah 21589
Saudi Arabia
http://spc.kau.edu.sa

■ Copy Price ■

- Local : SR 10.00
- External : US$ 10.00

■ Subscription ■

Scientific Publishing Centre, King Abdulaziz University
P.O. Box 80200, Jeddah 21589, Saudi Arabia

■ Exchange ■

Deanship of Library Affairs, King Abdulaziz University
P.O. Box 80213, Jeddah 21589, Saudi Arabia

(English Section)

(English Section)

JKAU: Comp. IT., **Vol. 2**, pp: 3 – 25 (2013 A.D./ 1435 A.H.)

DOI: 10.4197 / Comp. 2 -1

Protecting User Privacy from Social Networking Applications

Ebtesam A. Alomari, Maram S. Alhafizy, Omar A. Batarfi

Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, P.O.Box. 80221, Jeddah 21589, Kingdom of Saudi Arabia
obatarfi@kau.edu.sa

Abstract. Day by day, users of social network are increasing. They use different applications offered by social networking and publishing more personal information. However, users try to protect their privacy by keeping their information as private or share them only with family members or friends group. Further, many of them are unaware of the degree to which their privacy depends on third party applications and their developers, which left their private information vulnerable to accidental or malicious leaks by these applications. In this work, we focus on users' privacy concerns toward third-party application. We provide methodology to prevent these applications from sharing users' data with friends inside a social network without user permission, by the proposed system to allow the users through the installation process to give different level of permission to the data that is requested by friends' applications. Finally, the proposed system is evaluated for feasibility with theoretical analysis as well as simulation experiments by developing a client server application by Java to simulate a social network environment. Based on testing results, the proposed system assists users to have more control on their data and can achieve the level of privacy that they want.

Keywords: social network, third party application, privacy, security, and friends' applications.

1. Introduction

Social networking services are dominant on the web today, like Facebook, and LinkedIn. These services cater for a broad range of users with different ages, and backgrounds. They allow the users who have limited technical skills to publish personal information and communicate in an easy manner. People have different motivation to join social networks. They may create a profile for either professional or personal purposes, use the different applications offered by social network and share information with selected contacts or the public [1]. More and more user data are being published which may contain personal information that users may keep as private or share only with family members or friends group. Therefore, preserving user privacy in social network has become an important concern [2].

Social networking applications that facilitate online social interaction and sharing information among a large number of users are increasingly becoming a major type of online applications. These complex environments bring new security and privacy challenges [3]. Users constantly provide information to these applications, which may include privacy-sensitive information.

The main goal of the social network is to encourage the users to expand their social connectivity through interactions and content sharing with each other. Some social networks such as Facebook and MySpace provide some privacy control settings to their users, but some settings like the access and privacy control features are still limited and not flexible and robust, and not appropriate with a huge user base and high volume of privacy-sensitive content [3].

Third-party applications interact with a social network without being part of that social network. Social networking gives these third-party applications access to user data to provide some attractive services; however this may lead to serious privacy risks by exposing user data to these applications. To prevent privacy violations, social networking sites provide users with access control settings to place restrictions on who may view their personal information [4].

Disclosure of information about a user to other members that are not explicitly trusted by the user, without the permission of the user, has to be prevented. Privacy is the possibility to hide any information about any user, unless explicitly disclosed by the users themselves [1]. There are many researches on social networks for producing effective means of ensuring user privacy; most of these researches focused on techniques to prevent identity disclosure and neglected attribute disclosure attacks [5].

This paper focuses on two problems related to user data privacy that may be violated by third party applications in social networks, which are considered as non-trusted side, that may access to user information and share them without any restriction to protect user privacy. The major purpose of this research is to provide sufficient techniques which can secure data from others, protect to access data from other applications, and finally provide users more control over data to not be disclosed to the third-party applications.

2. Literature Review

Heng *et al.* [6] focused on users' privacy concerns toward third-party application on Facebook. They proposed two designs of privacy authorization dialogues, to control the data accessed and used by applications before adding them to the user's Facebook profile. They provided these solutions to address the defect in the current application installation dialogue in Facebook, which allow asking access request to group of different types of information together, such as asking access to all profile information while user may have different preferences for disclosing these different types of data. Therefore, they proposed two-design principles. The first principle indicated that privacy authentication dialogue should provide options for a user to give the permission for each data asked by the application. The second design was alerting user when applications ask for the sensitive private information by providing alert signals. However, there are two limitations in these solutions as we present in Fig.1. Their second proposed design for alert signal is not sufficient. Therefore, users may give permission to the application to share with others their data that they decided it before in privacy setting as private. This means that users may violate their privacy by themselves. In addition, as shown in Fig.1 application may get private information as necessary data, so user may not get attention to that and agree to complete installation process. However, we consider in this paper these two limitations and propose solution to restrict violating users' privacy and prevent disclosing their sensitive private data without permission.

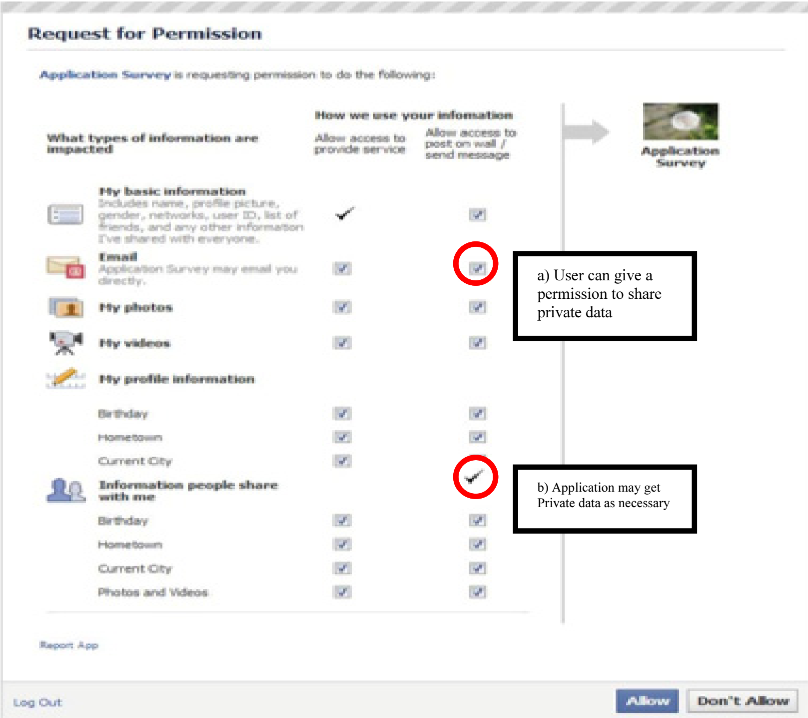


Fig. 1. Limitations in proposed solution by Heng *et al.*[6].

Moreover, Shehab *et al.* [7] illustrated an access control framework to control user data attributes shared with the third-party applications, based on enabling the users to determine the degree of specificity for the shared attributes. They supposed that services of applications are divided into four levels, base on the amount of services that they provide them for the users. These levels are minimum service, intermediate I, intermediate II, and maximum services. Further, they proposed that application informs the user the minimum requirement for each level of services, and then the social network asks users to select the level in which they prefer the application to function. However, in their solution users are not known whether this application need to get access to these data only or use them for share with others. Therefore in our solution, we consider this limitation and suppose levels of permission for each data allowed to the third-parity application, which means that user can decide which data he allows the application to get and for either access only or access and share with allowed friends.

Furthermore, Viswanath *et al.* [8] discussed the problem of sharing personal information with user friends while application can receives data from user friends and forwards them to other friends without their knowledge or permission. They called this attack Kevin Bacon attack and they indicated that there is a need to restrict information flow to protect against this problem. Further, they suggested executing the user principal in two sandboxes. The first one called read-only sandbox, which can read information shared from friends, but cannot write to the user database or share to other friends. The second sandbox can write to the user database and share with friends, but cannot read data shared from friends, which means that the data come from a friend is separated to prevent the Kevin Bacon attack. However, some applications need access to friends data to provide services. In this case our solution indicates that user will be able to decide which data accessed by friends application. Therefore, applications are not able to share user sensitive private data to others without user permission.

3. Problems

Third-party applications in social network are not considered as trustworthy, where they may access to users information and share them without any restriction to protect user privacy. This section discusses two problems related to user data privacy.

3.1 Sharing users data with their friends without permission

A large number of third-party applications needs sharing data with friend as presented by Viswanath *et al.* [8]. They analyzed 50 Facebook applications, including the top 25 applications and the rest were selected randomly from the top 1000 applications. Then, they examined each application whether it shares the data of the user with others or not. Therefore, they found the majority of applications about 52% require sharing data with friends only. This is considered the best behavior in terms of privacy for data sharing. However, this opens the possibility for a new kind of attack when application silently gets information about user and forwards it to a set of friends without user permission.

In social network today, when user wants to install the application, the authentication dialogue will appear to notify users about data that this application want to gets, as shown in Fig. 2.

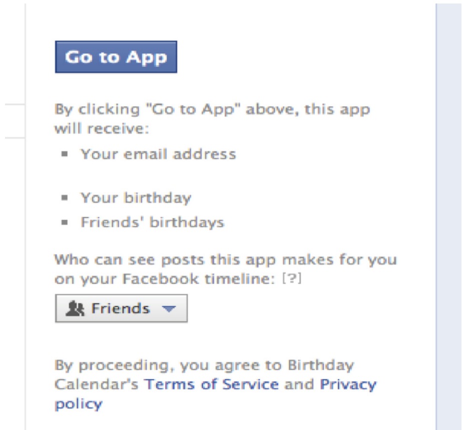


Fig. 2. Current authorization windows in Facebook.

However, during this process users do not have any control to prevent the applications from access to specific information or restrict them from sharing with others in social network. Moreover, the proposed solution by Heng *et al.* [6]. did not prevent application from access to user sensitive private data as presented in section 2.

3.2 Friends' applications get access to user private data

The current social network, such as Facebook, gives user the ability to determine data that are accessed by friends' application as shown in Fig. 3. However, there are two limitations to this specification method: 1) It is applied to all friends' application. 2) It may allow friends' application to access user data not intended for them to perform their function. For example the application may need birthday and photos while user, as shown in Fig. 3, allows them to access by many other data.

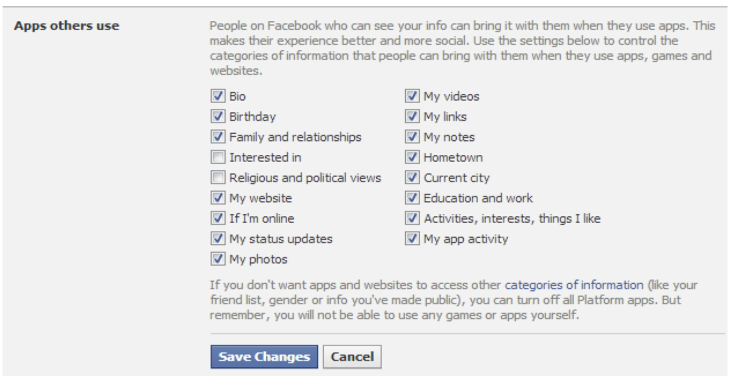


Fig. 3. The policies that apply in Facebook for friends' application.

4. Proposed Solution

Our contribution in this paper is proposing a methodology to counteract privacy loss due to access and share user's data by third-party applications in social networking. The solutions will take into consideration that the functionality of the applications is very important for users and the application developers need to get access to the data to provide services to them. Therefore, we suggest giving the decision for the user to decide the level of permission allowed to these applications. We propose a more efficient control model that allows users to control their data and to protect their privacy with the help of social network, as a trusted party.

Further, in the architecture of our solution, as shown in Fig. 4, the user can control the access rights for applications to his data by giving the permission is stored in social network. Therefore, through access or share request from application to social network, the social network should check the policy that was assigned by the user, stored in policy database, to know the level of permission allowed to this application. So, accessing or sharing user data will not be complete successfully if the application does not have the permission.

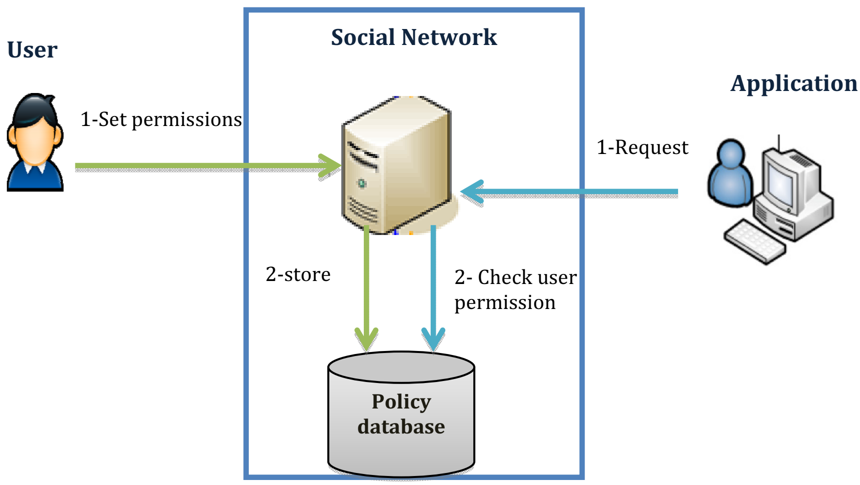


Fig. 4. System architecture.

4.1 Proposed Solution to Problem 1:

If a user does not want friends to know his private data, such as birthday and email, an application that has a permission to access these data should not be able to share them without permission. Therefore, we suggest that when a user wants to install an application, the authentication dialog containing the default setting asked by the application, will appear. It has a drop lists with three options:

- Allow access only: means application is allowed to access only to the data but it cannot share it inside the social network with the user friends.
- Allow access and share: means the application has the permission to access and share the data inside the social network with the user friends.
- Do not allow: means the application is prevented from access or sharing this data.

Accordingly the user could be able to change the permission assigned to data and chooses from the list as shown in Fig. 5. However, the user cannot change the permission “Required Access and Share” that assigned by the application for the required data. Furthermore, to enhance users awareness and protect them from giving the application by mistake, the permission to share their private data with the public in social network, we suggest an improvement in the solution proposed by Heng *et al.* [6] to handle their limitations as the following:

- *Attribute that considered private should have red “i” mark beside it, as shown in Fig. 5, to inform user that it is private and giving permission for application to share it with others will conflict with the previous privacy setting. So, user in this case should change the permission from ‘share and access’ to ‘access only’ or select ‘don’t allow’ to prevent application from getting this private data. However, if user does not change the permission, another window will appear before installation completes to alert user, as shown in Fig. 6.*

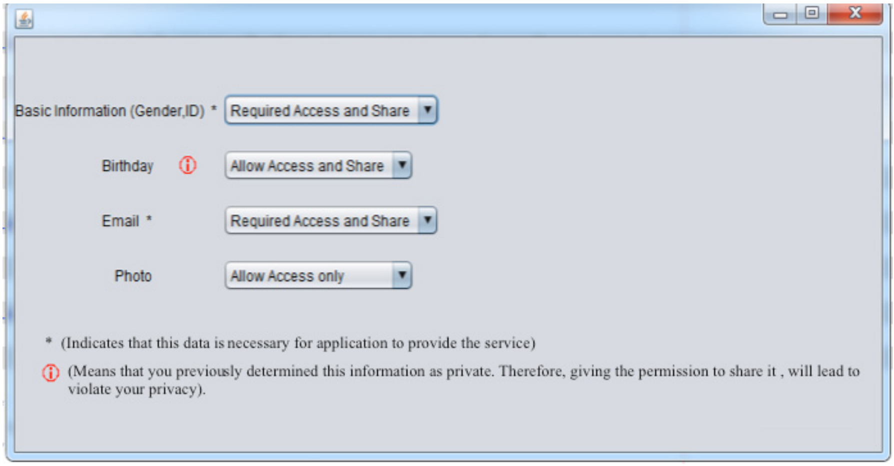


Fig. 5. Control access and sharing permission during installation.

Fig. 6. Alert users to change assigned permission.

- *When application needs a private attribute as a necessary to provide their service, an alert window should appear as shown in Fig. 7, to the user before the installation process complete. To inform user that this application will be able to violate the privacy by getting permission to share private data.*

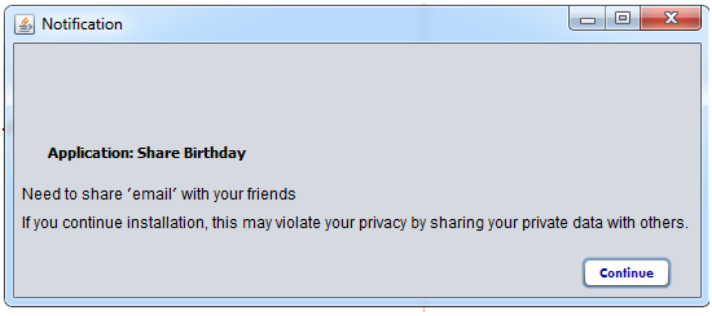


Fig. 7. Alert users during installation.

Therefore, as shown in Fig. 8, the sharing request by application will not perform if it does not have permission. As a result, the alert massages will enhance the users awareness toward their privacy.

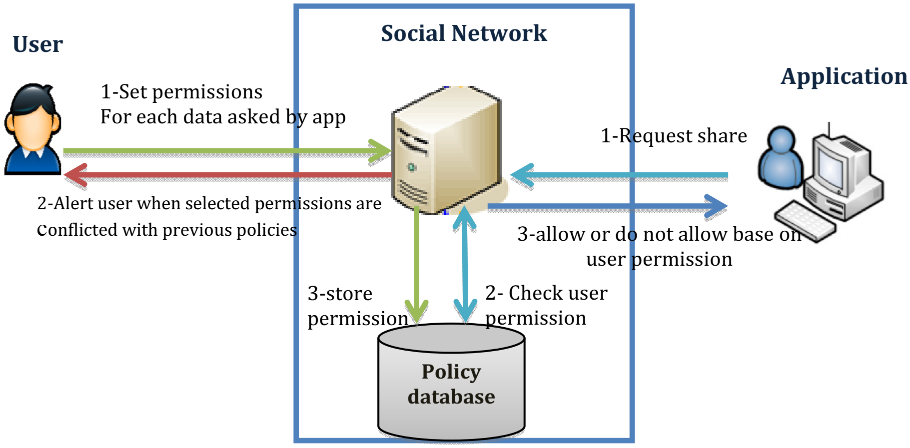


Fig. 8. Architecture for proposed solution-Part. 1.

As shown in Fig. 8 above, our proposed process from the user side is as follows:

1. User sets the permission for each application through the installation process and determines the level of access for each data.
2. Before the installation process complete, an alert message will be sent to user if there is a conflict with previous policies determined by users.
3. The determined permission by users for each data asked by each application will be stored in policy database of the social network.

Further, our proposed process when application intends to share user data with friends is as follows:

1. Application will send the request to social network to share user data with friends.
2. Social network will check the database to know whether user gives permission for sharing these data or not.
3. The social network will inform the application if it does not have a permission to share the data; otherwise the sharing request will perform successfully.

4.2 Proposed Solution to Problem 2:

For this problem, we propose an algorithm to add restrictions for accessing attributes by friends' applications and to address the limitation from specifying the general attributes for all friends' applications, as discussed in section 3. Therefore, we suggest the following:

1) The social network should give the user the ability to determine the level of accessing to the data for friends' applications, either level 1 or level 2.

- Level 1: means friends' applications can access only the attributes necessary to provide the service, and only if it matches the general attributes determined by the user previously.
- Level 2: means friends' applications can access any attributes that match the general attributes determined by the user previously.

2) The social network should give the user the ability to determine either to apply these levels of accessing for all friends' applications, or to determine the attributes that accessible by each friends' application individually. This technique will give the user more control of his information.

Further, in our scenario, we suppose that all application developers should inform the social network about the minimum attributes (necessary) for the application to provide the service. In addition, we suppose when a user registers at a social network, he/she decides either to give the same permission for all friends' applications or give the permission for each of them individually. If the user wants to give the permission of accessing to all friends' applications, then he/she should determine the general attributes to be accessed by friends' applications and the level of accessing as shown in Fig. 9.

4.3 Algorithm1 to access control on friends' applications

In the following algorithm the Social network: S is having:

Input : User set: $U=\{u_i \text{ where } i=1..\ .n\}$

Application set: $P=\{P_i \text{ where } i=1..\ .n\}$

User Data : A $=\{D_i \text{ where } i=1..\ .n\}$(age, birthday, photo)

Minimum attribute for application : Pi (mD1)

Boolean All_appction_access

General attributes access_by friends' applications: A ={GD1 , GD2, GD3}

Level of access L {Level 1 , level 2 }

Output : attributes if is allowed to access or notification

1) application P_i ⟶ requst_to_S_ to access_userAttribute (findI , Pi,D1,D2)

2) S check If All_application_access =true

{

3) if level =1

// check the minimum attribute for Pi match with general attributes

4) if MD1=GD1.

5) Then Allow(mD1)

6) else return " This attribute not allowed to access by friends' applications "

7)If level =2

//check is this attribute is general

8)Then Allow(GD1)

}

9)Else if

10) Massage appear to user from S select the attribute you want to access by this friends' application

11) Return Attribute that allowed to access by this friends' applications.

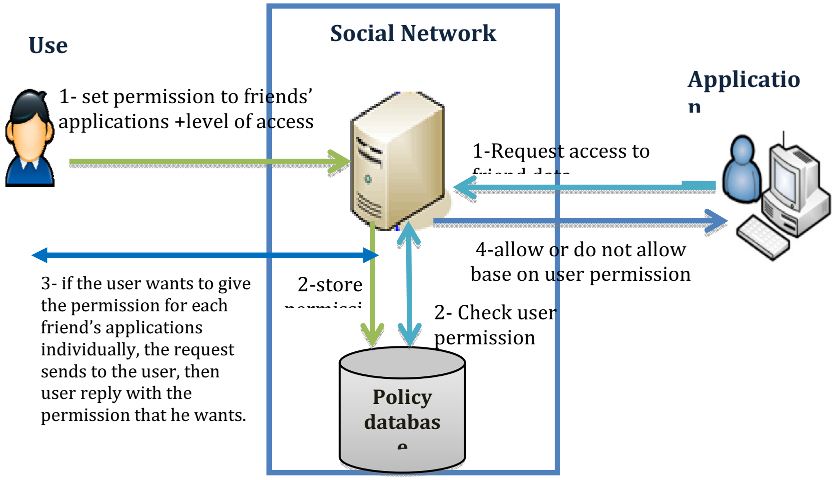


Fig. 9. Architecture for proposed solution – Part. 2.

As shown in Fig. 9 above, the proposed processes that the user should follow when he/she start registering in a social network are as follows:

1. User will determine the level to set the permission, as shown in Fig. 10.

Fig. 10. Determine the level of setting the permission for friends' applications.

1.1. Set the permission for all friends' application: means user will not specify different permission for each application. However, to protect user privacy, he/she should determine the following:

1.1.1. User determines general attributes to be accessing by friends' applications, as shown in Fig. 11.

Fig. 11. Determine general attributes to be accessed by friends' applications.

1.1.2. User gives restrictions in access control for friends' applications by giving the user two options, as shown in Fig. 12.

1.1.2.1. Allow friends' application accessing only necessary information if they match the policy determined before.

1.1.2.2. Allow friends' application accessing information determined before.

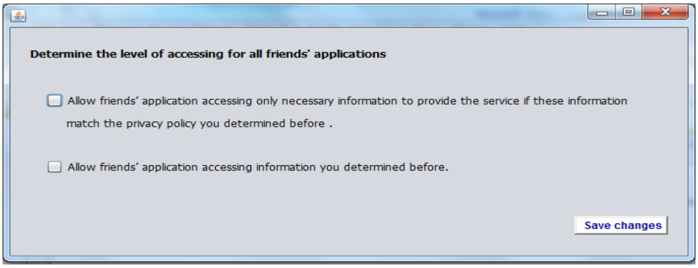


Fig. 12. Determine the level of accessing by friends' applications.

1.2. Set the permission for each friend's application individually as shown in Fig. 13. This gives the user ability to give the application permission to access only to attributes that necessary to provide the service. Therefore, the social network will notifies the user which attribute are necessary for this applications to provide the service.

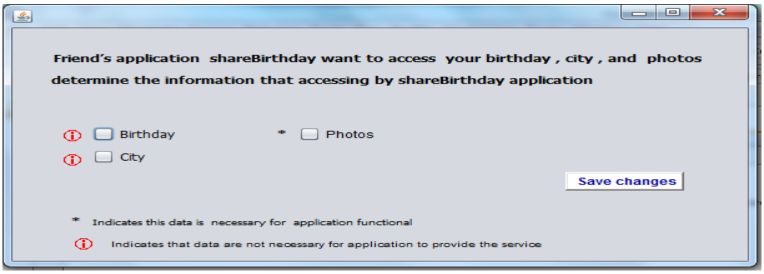


Fig. 13. Determine the attributes that accessing by each friends' application individually.

2. The determined permission by users will be stored in policy database of the social network.

In addition, when friends' application wants to access user data, the following process will be performed:

1. The friends' application send request to social network to access the user's data.
2. Social network checks policy database to know whether the user gives the permission for this application to access these data or not. If the user gives the permission for each friends' applications individually, go to step 3 then 4, else go to step 4 directly.
3. The request is sent to the user, then the user replies with the permission that he/she likes.

4. The social network will inform the application if it does not have a permission to access the data, otherwise the access request will be performed successfully.

5. Prototype Implementation

We evaluated the feasibility of our solution by simulating the social network environment through implementing a client server application by java language. Then, we connected the server with the database that we created to store data and policies for both the users and applications.

For problem.1, we simulate the social network control process of sharing data with friends under users policies. As shown in Fig.14, when application requests sharing user data with friends, the "Check_Sharing_Permission" method will be called and received the user id, application id, friend list and the data that app request to share them with user's friends. Therefore, server (social network) will check the database that contains the user policies to allow or prevent sharing request. If application has a sharing permission, the sharing process with allowed friend will perform successfully. Otherwise, the application will not be able to share this data because user does not give permission to this application to access or share.

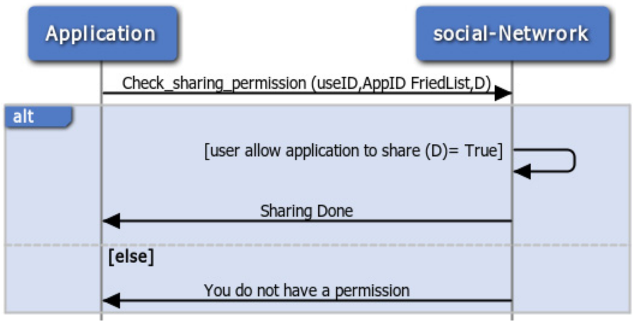


Fig. 14. Sequence of sharing data with friends.

For problem.2, we simulate the social network access control process for accessing data by fiend's applications under users policy after we implemented the algorithm. 1.

When a friend's application intends to access some attribute for users, it will send a request to a social network by calling "request_access_userAttribute" method, which informs the social network the user id, application id, and the data that app request to access them, as shown in Fig. 15. Then the social network checks whether the user sets the permission to all friends' applications or not.

- If user gives the permission to each friends' application individually, a user will receive notification from social network by calling "request_access_attribute" method, that provides the user with the id of this application, data that application needs to access, and which attributes are necessary for this application to become functional, as shown in Fig. 15. Then the user determines the attribute to allow for this application to access, and send the permission to social network, which allow to application to access attribute depended in this permission.
- If the user gives the permission to all friends' applications at once. Then the social network checks the level of accessing. If it is level 1, the social network knows that the user allows only the minimum attribute of the application. If it is level 2, social network knows that the user allows the general attributes that defined already to access with friends' applications and allow the application to access permitted attributes.

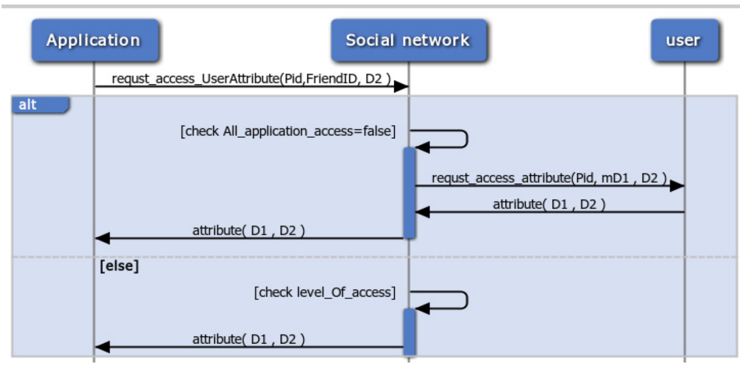


Fig. 15. Sequence diagram for Algorithm. 1.

6. Results and Discussion

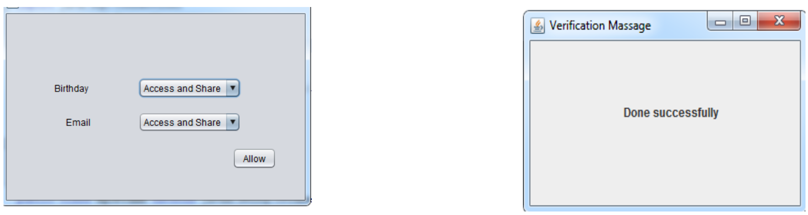
In this section, we focus on the evaluation of our hypothesis, to know whether they are valid or not. In our client-server application we run both applications where the user considers as client and the server act as social network that have the database. Then, we test different users inputs in order to check their effect on application side. We provided the print screens for both user and application.

6.1 For problem 1

We tried to test two cases for two users. We suppose both of them installed application called "share birthday" which needs two attributes (birthday and email) to be accessed. Further, we consider that the users want to use this application to share their birthday with friends so both of them give the application permission to "access and share" their birthday. However, for 'email' attribute, user.1 allows access whereas user.2 allows application to access and share with friends as shown in Fig.16(a), Fig.16(b) and Table 1. For user.2 when application invokes Check_sharing_permission method, which is explained in section 4, the social network checks the roles table in database and finds that the user gives a permission to share this attribute. So, sharing done successfully as shown in Fig.17(b). However, for user.1 sharing will not be allowed as shown in Fig. 16, because when the social network checks the roles table, it finds the application has a permission to access only, as shown in Fig.17(a).

(a) User.1 input to 'Roles database' (b) output massage for application

Fig. 16. Result for user.1 test.



(a) User.2 input to 'Roles database' (b) output massage for application related to user.2

Fig. 17. Result for user.2 test.

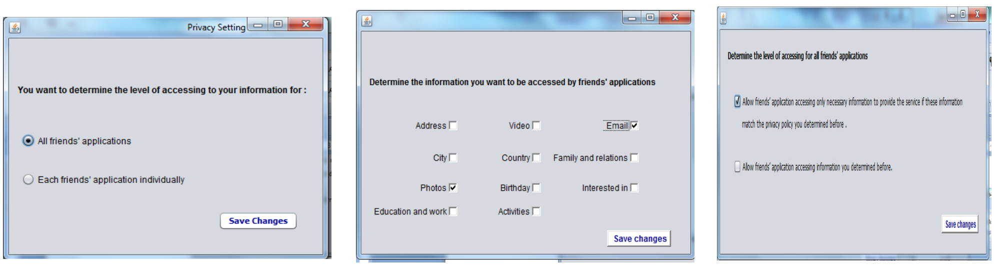
Table 1. Users level of permission for ShareBirthday application.

User number	Name of attribute (D)	Level of permission	Social Network - process
1	'Birthday'	Access and share	Allow sharing
2	'Birthday'	Access and share	Allow sharing
1	'Email'	Access and share	Allow sharing
2	'Email'	Access only	Do not Allow sharing

Therefore, we found that these techniques will increase users control on data to protect privacy and empower users to prevent sharing data with friends without permission. This means that the possibility of violating their privacy by a third-party application decreases.

6.2 For problem 2

We suppose there are three users, and an application called "share photos" needs an email and photos of the user, while its minimum attribute is photos. For user.1 he/she gives the permission to all friends' application to access only the minimum attributes it needs when they match with general attributes as shown in Fig.18 (a, c). The user sets the general attributes as follows: email and photos, as shown in Fig.18(b). When "share photos" application requests to access user.1 data to test the permission for photos and email attributes, the result is the photos access by "share photos" application as shown in Fig.19(a).



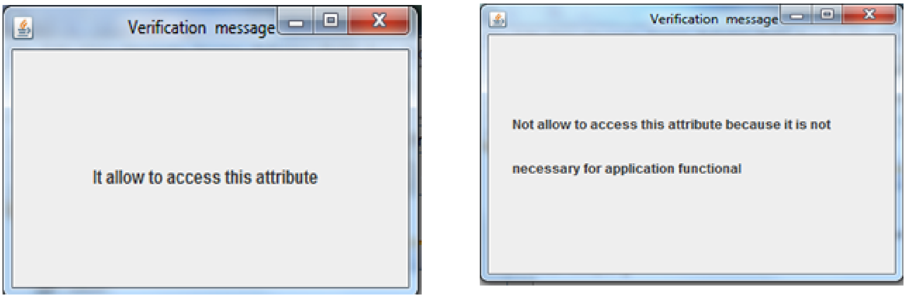
A) User.1 determines the level of setting the permission for friends' applications

b) User.1 determines general attributes to be accessed by

c) User.1 determines the level of accessing by friends' applications

Fig.18 User.1 input to 'Roles database'.

Because user.1 determines the minimum attributes should be accessible by friends' application, when they match the general attributes, the application cannot access to email, because it is not necessary for its function, as shown in Fig.19 (b).

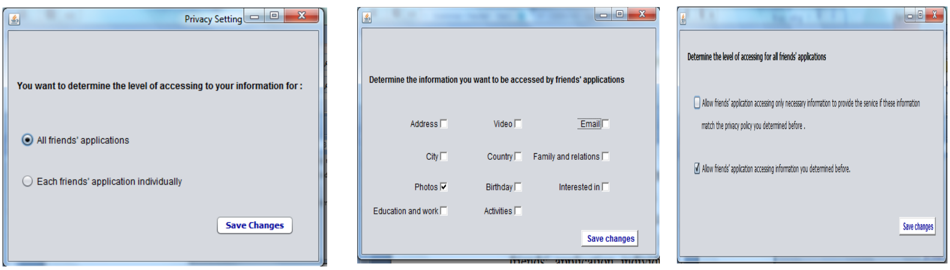


a) The result for test user.1 photos attribute

b) The result for test user.1 email attribute

Fig. 19. Result for user.1 test.

User.2 gives the permission to all friends' applications to access the general attributes and determines the photos as general attributes as shown in Fig.20. The result is the photos access by "share photos" application, but the application cannot access to email because user.2 doesn't determine it as general attributes, as shown in Fig.21.

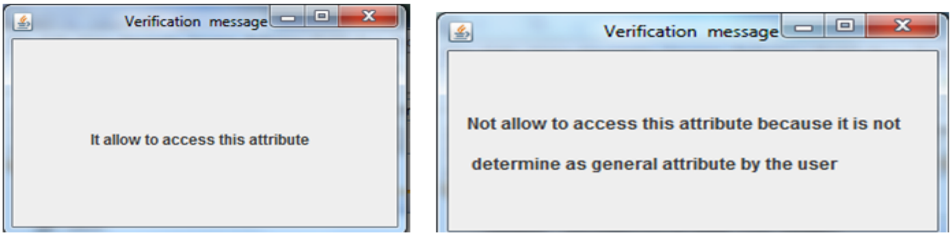


A) User.2 determines the level of setting the permission for friends' applications

b) User.2 determines general attributes to be accessed by friends' applications

c) User.2 determines the level of accessing by friends' applications

Fig. 20. User.2 input to 'Roles database'.

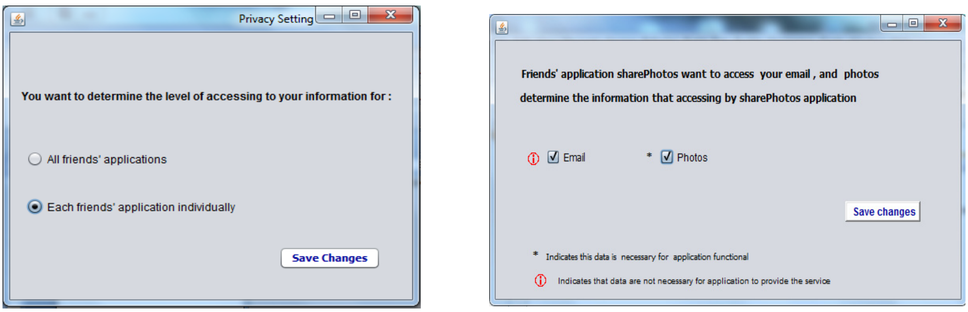


a) The result for test user.2 photos attribute

b) The result for test user.2 email attribute

Fig. 21. Result for user.2 test.

User.3 wants to determine the level of accessing for each friends' application individually as shown in Fig.22(a). For example, “Share Photos” application want to access user.3 photos, email. The result will depends on the user’s permission as shown in Fig.22(b). Table.2 summarizes these three cases. These results indicate that our hypothesis is correct.



a) User.3 determines the level of setting the permission for friends' applications

b) User.3 determines attributes to be accessed by share Photos applications

Fig. 22. Result for user.3 test.

Table 3. Summary of three cases.

User ID	Permission to all friends' application	Level of accessing by friends' application	General attributes to be access by friends' application	Minimum attribute for "share photos" application	Unnecessary attribute for share photo application"	Accessing attribute by "share photos" application
1	True	Level 1	Email, photos	Photos	Email	Photos
2	True	Level 2	Photos	Photos	Email	Photos
3	False	-	-	Photos	Email	Photos, email

7. Conclusion

Recently, the concern about user privacy in social network is raised, because the number of users who use the online social network to share personal and private information is increased. Furthermore, third-party applications in social network are becoming a major type of online applications as they provide more attractive services for users. However, we cannot trust those applications developers since their intents are not known. Therefore, preserving user privacy in social network from third party applications has become an important concern. This paper focused on this issue and discussed two problems, which are, how could users avoid sharing their private sensitive data with their friends without their permission, and how could they customize the access of friend's applications to their data.

To avoid leaking personal information to friends, the authors suggest giving users, through installation process, more ability to specify level of permission for each data asked by third-party application. In addition, to give users control on friend's application, the authors propose an algorithm to enable them determining the attributes that can be accessed. This determination can be for all friends' applications at once or for each application individually. Moreover, to evaluate the feasibility of the proposed solution, the authors simulate the environment of social network by building client server application by java language. Then, they connected it with a database containing applications and users data and policies.

We found that when user applies our approach, he/she can change the permission for each attributes and become able to achieve for his data the level of privacy that he/she wants. Therefore, the researchers found that their proposed solutions help users to control their data given to third-party application, especially the sensitive and private information that they usually keep private or visible only to specific group. Overall, this work focuses on preventing third-party application from sharing user's data with friends in addition to avoiding friend applications from getting data without permission. However, there is still a need for future researches to find how to prevent third party applications from exposing publicly or to advertisers, the information that we allow to them.

References

[1] **Cutillo, L. A., Molva, R.** and **Strufe, T.,** "Safebook: A privacy-preserving online social network leveraging on real-life trust", *Communications Magazine, IEEE*, 47.12: 94-101, (2009).

[2] **Tripathy, B. K.** and **Panda, G. K.,** "A new approach to manage security against neighborhood attacks in social networks", *Advances in Social Networks Analysis and Mining (ASONAM)*, *International Conference on. IEEE*, (2010).

[3] **Masoumzadeh, A.** and **Joshi, J.,** "Osnac: An ontology-based access control model for social networking systems." *Social Computing (SocialCom), IEEE Second International Conference on. IEEE*, (2010).

[4] **Adrienne, F.** and **Evans, D.,** "Privacy protection for social networking APIs", *Web 2.0 Security and Privacy* (W2SP'08) (2008).

[5] **Sean, C.** and **Srivastava, G.,** "Social network privacy for attribute disclosure attacks." *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, (2011).

[6] **Heng, X., Wang, N.** and **Grossklags, J.,** "Privacy By Redesign: Alleviating Privacy Concerns For Third-Party Applications, (2012).

[7] **Shehab, M.,** ***et al.*,** "Access control for online social networks third party applications." *Computers & Security* (2012).

[8] **Bimal, V., Kiciman, E.** and **Saroiu, S.,** "Keeping information safe from social networking apps." *Proceedings of the 2012 ACM workshop on Workshop on online social networks*. ACM, (2012).

حماية خصوصية المستخدم من تطبيقات شبكات التواصل الاجتماعية

إبتسام العمري، ومرام الحافزي، وعمر باطرفي

قسم تقنية المعلومات

كلية الحاسبات وتقنية المعلومات

جامعة الملك عبدالعزيز، صندوق بريد ٨٠٢٢١، جدة ٢١٥٨٩،

المملكة العربية السعودية

obatarfi@kau.edu.sa

المستخلص. يومًا بعد يوم، يزيد عدد مستخدمي شبكة التواصل الاجتماعي وبالتالي يزيد استخدام التطبيقات المختلفة التي توفرها شبكات التواصل لنشر المعلومات الشخصية . علاوة على ذلك، فإن المستخدمين يحرصون على حماية خصوصياتهم ومشاركة هذه الخصوصيات مع أفراد العائلة أو الأصدقاء. ومع ذلك، فإن العديد منهم غير مدركين لدرجة هذه الخصوصية. وفي هذا البحث، سيتم استعراض خصوصيات المستخدمين لإنشاء التطبيقات، والتركيز على منهجية منع مشاركة الخصوصية بين الأصدقاء داخل شبكة التواصل دون إذن المستخدم. وأخيراً سيتم تطبيق آلية تقويم النظام المقترح لجدوى النظرية مع إجراء نظام محاكاة بلغة جافا في بيئة الخادم والعميل.

JKAU: Comp. IT., **Vol. 2**, pp: 27 - 44 (2013 A.D./ 1435 A.H.)

DOI: 10.4197 / Comp. 2-2

A Novel Approach of Selection Sort Algorithm with Parallel Computing and Dynamic Programing Concepts

Khaled Thabit and **Afnan Bawazir**

Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, P.O.Box. 80221, Jeddah 21589, Kingdom of Saudi Arabia.
afnan-bawazir@hotmail.com

Abstract. Many research works have been conducted to find out better enhancement for Selection Sort Algorithm, such as bidirectional selection sort "Friend Sort Algorithm" which can position two elements in each round. We have improved this algorithm by using the concept of parallel computing. This algorithm is called Min-Max Bidirectional Parallel Selection Sort (MMBPSS). Also this paper proposes to use dynamic programming (stack) to reduce sorting time by increasing the amount of space. The basic idea behind using stack is to eliminate unnecessary iteration. This algorithm is called Dynamic Selection Sort "DSS". To fuse advantages of "DSS" with advantages of "MMBPSS", we suggest a new third algorithm called Min-Max Bidirectional Parallel Dynamic Selection Sort "MMBPDSS". It can position two elements: minimum and maximum from two directions using Dynamic Selection Sort algorithm in each round in parallel, thus reducing the number of loop required for sorting. Results obtained after implementation are provided in graphical form with an objective to show that "MMBPDSS" is saving almost 50% of classical selection sorting time and ensure accuracy.

Keywords: *High Performance Computing, Selection Sort, Bidirectional selection sort, parallel computing, Dynamic Programming.*

1. Introduction

Sorting is a technique by which elements are arranged in a particular order following some characteristic or law[1]. Data can be in numerical or character form. There are a lot of sorting techniques, currently used in industrial applications and academic researches, to arrange the data of various forms and from different areas. Sorting is of considerable importance as the human is possessed in keeping the ordered information/knowledge. To search the information efficiently the arrangement of data is very important. To facilitate the human, computers consume a massive amount of time in ordering the data[2]. There are a lot of sorting algorithms used nowadays such as Bubble Sort, Insertion Sort, Selection Sort and Cocktail sort. Every kind of sort has its own pros and cons, and the pattern of input data is a major factor for its performance. This paper focuses on Selection Sort algorithm which has performance advantages over more complicated ones in certain situations, especially where auxiliary memory is limited. It does many comparisons and least amount of data swapping. Selection Sort algorithm is inefficient on large lists, because it has O (n2) complexity, and generally performs worse than the similar Insertion Sort. Many research works have been conducted to find out better enhancement for Selection Sort [1, 3-5] that speed up the sorting process such as bidirectional Selection Sort Algorithm, which can position two items in each pass thus reducing the number of loops required for sorting. This algorithm also called "Friends Sort"[3]. Lakra and Divy[4] suggested "Double Selection Sort" which makes sorting an efficient and convenient way for larger data set by saving almost 25% to 35% than the classic Selection Sort Algorithm. We have improved "Friends Sort" algorithm by making it working in parallel. This algorithm is called Min-Max Bidirectional Parallel Selection Sort (MMBPSS). Also other study proposes an improvement for Selection Sort Algorithm by using dynamic programming technique (Stack). The key idea behind using stack is to eliminate unnecessary passes by reducing the number of comparison. A Stack is used to store the location of previous max element found, and instead of starting from the beginning each time, the largest element is found and placed at the end of the array. This algorithm is called Dynamic Selection Sort "DSS". We suggest a new third algorithm called Min-Max Bidirectional Parallel Dynamic Selection Sort "MMBPDSS" which combine DSS and MMBPSS. Our hypothesis "MMBPDSS" makes sorting an efficient and convenient way for smaller and larger data set by saving almost 50% than the classic Selection Sort and Friend Sort algorithms [3] due to the parallel implementation of the algorithm.

The paper is organized as follows: a brief review of selection sorting algorithm are discussed in Section 2, while section 3 contains the proposed algorithm "Min-Max Bidirectional Parallel Selection Sort" and explained it in more details, the steps, and procedure with an example. Section 4 presents second proposed algorithm “Dynamic Selection Sort” in details with procedures using example. Furthermore, Section 5 contains third proposed algorithm “Min-Max Bidirectional Parallel Dynamic Selection Sort” in details with steps and procedure using example. This paper further progress in Section 6 by testing and analyzing the proposed algorithm’s results with the classical Selection Sorting and the new Friend Sorting technique[3]. Finally the paper concluded in Section-7.

2. Brief Review of Selection Sorting Algorithm

This section presents a review of Selection Sort including history of formation, methodologies and algorithms.

2.1 Selection Sort

Selection Sort is a well-known sorting technique that scans an array to find the maximum item, puts it at the last location in the array, and then scans the array for the second maximum item, puts it before the last location, then third maximum and so forth, until reaches the smallest item to be put at the first location of the array. It has $O(n^2)$ complexity, inefficient for larger lists or arrays and its performance is worse than that of Insertion Sort. In certain situations, it has a prominent efficiency than some other convoluted algorithms. The number of passes, of the Selection Sort for a given list, is equal to the number of elements in that list.[6] The number of interchanges and assignments depends on the original order of the items in the list/array, but the sum of these operations does not exceed a factor of n^2[7].

2.2 Procedure for Selection Sort

```
Procedure Selection-Sort (List: List of items to be sorted)
      Length ← length (List);
      For i ← Length -1 to 1 do
              Max ← i;
              For j ← i - 1 to 0 do
                      if( List[ j ] > List[ Max ] )
                              Max ← j
                      End if
```

```
                End for
                Swap (List[i], List [Max]);
    End for
    End Procedure
```

2.3 Min-Max Bidirectional Parallel Selection Sort

Min-Max Bidirectional Parallel Selection Sort (MMBPSS) is an improvement on the idea of traditional Bidirectional Selection Sort and Friend Sort Algorithms[3] which can position two elements in each round parallel, thus reduces the number of loop required for sorting. The basic design idea of the (MMBPSS) is as follows: it divides the list into two parts, minimum and maximum values from each part are selected in each sort round. Then both values of minimum and maximum from each part are compared to determine the minimum and the maximum of the whole array, and they are placed at their proper locations. The Steps of the proposed algorithm are as follows:

1. Divide the array into two.

Now: working in parallel from 2 to 7:

2. Find minimum and maximum values from each part.
3. Take minimum1 of the first part, compare it with minimum2 of the second part.
4. Swap and put them at their exact location.
5. Take maximum1 of the first part, compare it with maximum2 of the second part.
6. Swap and put them at their exact location.
7. Repeat these steps for the whole array.

2.4 Procedure for MMBPSS

```
Procedure MMBPSS (List: List of items to be sorted)
        Length ← length (List);
        Max, Min;
Mid = Length/2;
Start = 0, end = Length-1;

                For i← Start to end do in parallel
                        () =>
```

```
For j← Start to mid-1
        if( List[ j ]  <  List[ Min1 ] )
                Min1 ← j
        End if
        if( List[ j ]  >  List[ Max1 ] )
                Max1 ← End
        End if
    End for
        () =>
For K← mid to end
        if( List[ k ]  <  List[ Min2 ] )
                Min2 ← k
        End if
        if( List[ k ]  >  List[ Max ] )
                Max2 ← k
        End if
If (List [max1] >= List [max2])
        Max = max1;

        Else
          Max = max2;
End if
Swap (List[i], List [max]);

        If (List [min1] <= List [min2])
        Min = min1;
        Else
        Min = min2;
End if  Swap (List[ i ], List [ min ] );
        End for
End Procedure
```

2.5 Example for MMBPSS

Let us take an array as an example see (Figure. 1) to apply (MMBPSS) on it:

5	33	8	41	19	2	50	1	7	20

Fig. 1. Unsorted Array

Index of Mid=5

Each for loop work at one part to find min & max in parallel, See (Figure. 2)

5	33	8	41	19	2	50	1	7	20

Fig. 2. Divide Array into two parts.

First iteration: see (Figure. 3)

Min1= 5 Min2=1

Max1=41 Max2=50

Then compare them:

Min 1> Min 2 → Min= Min 2= 1

Max 1< Max 2 → Max= Max2 = 50

1	33	8	41	19	2	20	5	7	50

Fig. 3. Array after the first iteration.

Second iteration: see (Figure. 4)

Min1= 8 Min2=2

Max1=41 Max2=20

Then compare them:

Min 1> Min 2 → Min= Min 2= 2

Max 1> Max 2 → Max= Max1 = 41

1	2	8	7	19	33	20	5	41	50

Fig. 4. Array after the second iteration.

Third iteration: see (Figure. 5)

Min1= 7 Min2=5

Max1=19 Max2=33

Then compare them:

Min 1> Min 2 → Min= Min 2= 5

Max 1< Max 2 → Max= Max2 = 33

1	2	5	7	19	8	20	33	41	50

Fig. 5. Array after third iteration.

Fourth iteration: see (Figure. 6)

Min1= 7 Min2=8

Max1=19 Max2=20

Then compare them:

Min 1< Min 2 → Min= Min 1= 7

Max 1< Max 2 → Max= Max2 = 20

1	2	5	7	19	8	20	33	41	50

Fig. 6. Array after fourth iteration.

Fifth iteration: see (Figure. 7)

Min1= 19 Min2=8

Max1=19 Max2=8

Then compare them:

Min 1> Min 2 → Min= Min 2= 8

Max 1> Max 2 → Max= Max1 = 19

1	2	5	7	8	19	20	33	41	50

Fig. 7. Array after fifth iteration.

Finally, the array is sorted. See (Figure. 7)

2.6 Dynamic Selection Sort

Dynamic Selection Sort (DSS) is an improvement on the idea of Selection Sort but it used dynamic programming (stack) to reduce sorting time by increasing the amount of space. The basic idea behind using stack is to eliminate unnecessary iterations. A stack is used to store the location of previous largest element found, instead of starting from the beginning after the largest element is found and placed at the end of the array, we pop the stack and start at the location of the previous max, so we can decrease the number of comparison required for sorting operation. Here we used two Stacks, one to store the location of the previous index largest element, another one to store value. Once the location and value of the previous largest element is popped off from the stack and compared with the elements of the array from that location till a new largest element is found or the new end of the array is reached. If a larger element is encountered then the location and value of the previous largest element is pushed into the stack. The new largest element is again compared to the remaining elements of the array. This process is repeated until the array is sorted.

2.7 Procedure for DSS

Procedure DynamicSelectionSort (List: List of items to be sorted)

```
Length ← length (List); Max, Location, Value, Stack1, Stack2;
For i ← Length -1 to 1 do
        Max ← i;
        For j ← i - 1 to 0 do
                if( List[ j ] > List[ Max ] )
                        Max ← j
                        Push Max in Stack1
                        Value ← List [Max]
                        Push Value in Stack 2
                End if
        End for
        Swap (List[ i ], List[Max] );
        Pop the first element from both the stack //this element
already has been swapped
        While (Stacks are not empty && i > 0)
                i ← i - 1
                Location ← pop element from Stack1, Max ←
Location
                Value ← pop element from Stack2.
                Swapped ← false
```

```
                For n ← Location - 1 to 0 && i - 1 do
                        Swapped ← true;
                        if ( List[ n ]  > Value )
                                Max ← n
                                Push Max in Stack1
                                Value ← List [Max]
                                Push Value in Stack 2
                        End if
                End for
                If (Swapped )
                        Swap (List[ i ], List[Max] )
                        Pop the first element from both the stack
                Else
                        i ← i + 1
        End while // Stack count loop
    End for // outer for loop
```

End Procedure

2.8 Example for DSS

Let us take an array as an example (Fig.8 and 9) to apply (DSS) on it:

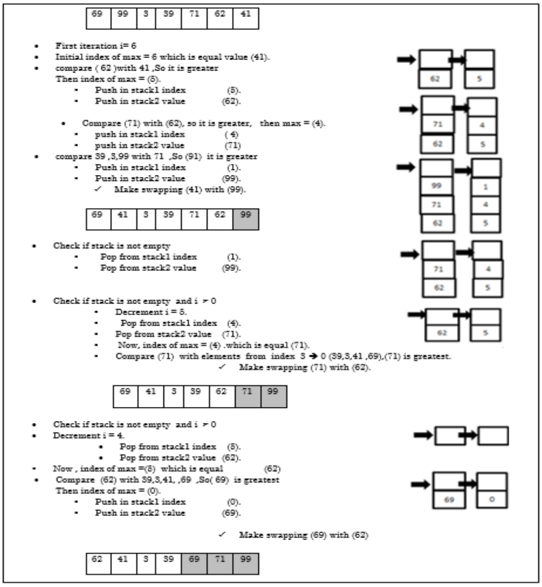


Fig. 8. DSS algorithm by way of an example.

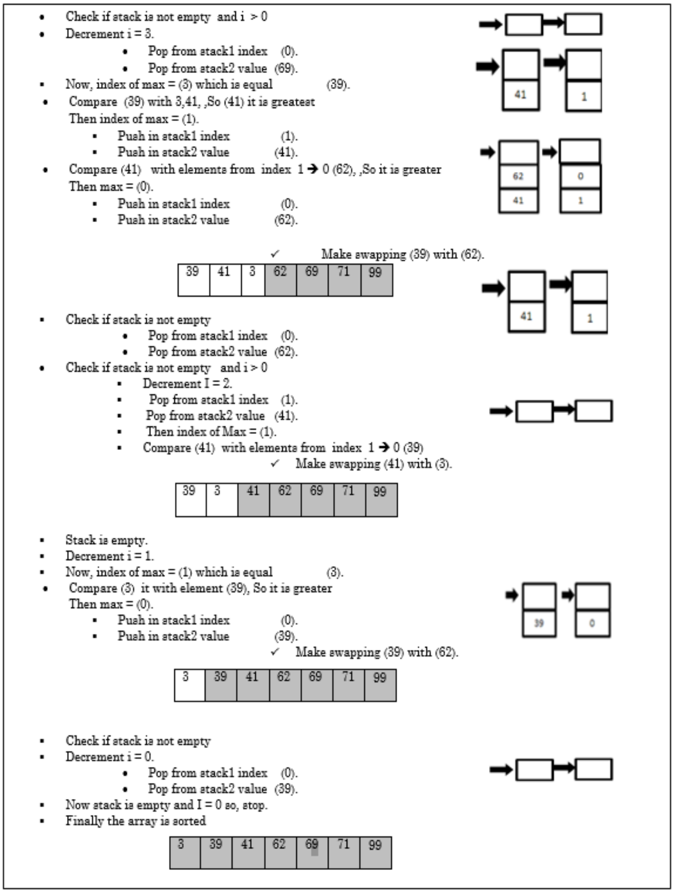


Fig. 9. DSS algorithm by way of an example "cont".

3 Min Max Bidirectional Parallel Dynamic Selection Sort

Min-Max Bidirectional Parallel Dynamic Selection Sort (MMBPDSS) is an improvement on the idea of Dynamic Selection Sort which can position two elements, minimum and maximum, from two directions in each round in parallel. It thus reduces the number of loop required for sorting. The Steps of the (MMBPDSS) are as follows

1. First thread starts from the beginning of the array which finds the smallest element (using 2 stacks to store the minimum and its location).

2. The second thread starts from the end of the array searching for the largest element (using 2 stacks to store the maximum and its location).

3. Then concatenate the first half from first thread with the second half from the second thread.

3.1 Procedure for MMBPDSS

ProcedureMinMaxBidirectionalParallelDynamicSelectionSort (List: List of items to be sorted)

First Thread: // sorting smallest elements

```
Length ← length (List); Min, Location, Value, Stack1, Stack2, mid = Length
/ 2;
        For i ← 0 to mid-1 do
                Min ← i;
                For j ← i - 1 to 0 do
                        if( List[ j ]  < List[ Min ] )
                                Min ← j
                                Push Min in Stack1
                                Value ← List [Min]
                                Push Value in Stack 2
                        End if
                End for
                Swap (List[i], List [Min]);
                Pop the first element from both the stack //this element
already has been swapped
                While (Stacks are not empty && i <= mid)
                        i ← i + 1
                        Location ← pop element from Stack1, Min← Location
                        Value ← pop element from Stack2.
                        Swapped ← false
                        For n ← Location - 1 to 0 && i - 1 do
                                Swapped ← true;
                                If (List[n] < Value)
                                        Min ← n
                                        Push Min in Stack1
                                        Value ← List [Min]
                                        Push Value in Stack 2
                                End if
                        End for
                        If (Swapped)
                                Swap (List[i], List [Min])
```

```
                Pop the first element from both the stack
            Else
                i ← i - 1
        End while // Stack count loop
    End for // outer for loop
End first thread.
```

Second Thread: // sorting largest elements

```
Length ← length (List); Max, Location, Value, Stack1, Stack2, mid = Length / 2;
    For i ← mid to 1 do
        Max ← i;
        For j ← i - 1 to 0 do
            if( List[ j ] > List[ Max ] )
                Max ← j
                Push Max in Stack1
                Value ← List [Max]
                Push Value in Stack 2
            End if
        End for
        Swap (List[i], List [Max]);
        Pop the first element from both the stack //this element already has been swapped
        While (Stacks are not empty && i >= mid)
            i ← i - 1
            Location ← pop element from Stack1, Max ← Location
            Value ← pop element from Stack2.
            Swapped ← false
            For n ← Location - 1 to 0 && i - 1 do
                Swapped ← true;
                If (List[n] > Value)
                    Max ← n
                    Push Max in Stack1
                    Value ← List [Max]
                    Push Value in Stack 2
                End if
            End for
            If (Swapped)
                Swap (List[i], List [Max])
                Pop the first element from both the stack
            Else
                i ← i + 1
        End while // Stack count loop
    End for // outer for loop
```

End second thread.
End Procedure.

3.2 Example for MMBPDSS

Let us take an array as an example (Figure. 10) to apply (MMBPDSS) on it:

2	7	10	6	1	8	9	12	11	7	3	4

Working in parallel
First thread sorting smallest element using Dynamic Selection Sort

1	2	3	4	6	7	9	12	11	7	10	8

Second thread sorting largest element using Dynamic Selection Sort

2	7	6	4	1	3	7	8	9	10	11	12

Then concatenate first half form first thread with second half from second thread.

1	2	3	4	6	7	7	8	9	10	11	12

4. RESULTS AND DISCUSSION

To prove efficiency of MMBPSS, DSS and MMBPDSS performance, they were implemented along with Classical Selection Sort algorithm and with the new Friend Sorting Algorithm[3]. The calculation of average execution time, total comparison and swapping frequencies are conducted for random sample lists with different sizes, 30 times for all mentioned algorithms in the paper. We have conducted those algorithms by using basic laptop with the following specification Intel Core2Duo processor 2.53 GHZ machine with 4 GB.

The results are accomplished in three ways:
1. Comparison of execution time of MMBPSS, DSS and MMBPDSS algorithms with the classic Selection Sort algorithm and with the new Friend Sorting algorithm.
2. Comparison of total frequency of MMBPSS, DSS and MMBPDSS with the classic Selection Sort algorithm and new Friend Sorting algorithm.
3. Comparison of total swapping frequency of MMBPSS, DSS and MMBPDSS with the classic Selection Sort algorithm and with the new friend sorting algorithm.

4.1 Comparison of Execution Time

Table 1 shows the comparison of the MMBPSS, DSS and MMBPDSS algorithm with the classic Selection Sort algorithm and with the new Friend Sorting algorithm with respect to the average execution time each algorithm takes to perform sorting.

Table 1. Time Comparison

Array size	Average of Execution Time (Test on different array of size)in millisecond				
	Selection Sort	New Friend Sorting Algorithm	Min-Max Bidirectional Parallel Selection Sort	Dynamic Selection Sort	Min-Max Bidirectional Parallel Dynamic Selection Sort
1000	13.44133333	11.99764	36.87646667	10.43051	11.25
3000	44	45	117	37	31
5000	206.7171	169.3317	192.08728	140.6054	87.25
7000	213.9861	195.2054	198.8224	164.0168	125.4608
10000	693.21615	665.3608	528.2458	394.7421	299.2226
30000	4224.902	4244.429	3630.884	3498.265	2657.741
50000	11916.28	12019.771	9281.721	8580.618	6537.066
70000	18795.62	19479.03	16521.67	14323.28	12189.42
100000	53993.07	52417.33	31804.16	35361.93	27735.59

Graphical view for Table.1 is presented in Figure. 11.

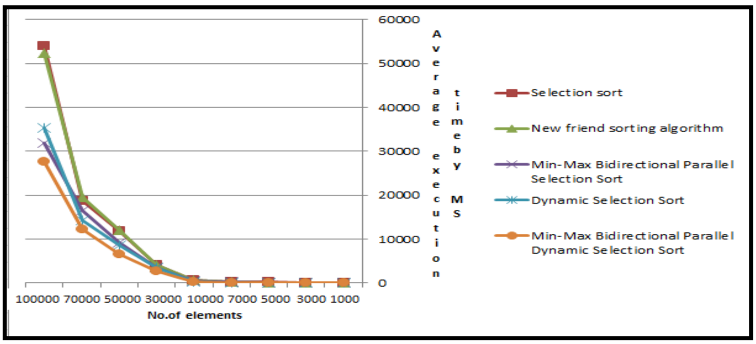


Fig. 11. Time Comparison.

It can be observed from Fig. 11 that the performance of the new Friend Sorting algorithm is less efficient when the array size is smaller than 30000 but after that its efficiency degrades and it is equally efficient to the classic Selection Sort but MMBPSS is more efficient when the array size is over 35000 elements. There is an additional overhead when applying MMBPSS on smaller array size. DDS reduces the execution time compared to the classic Selection Sort, the new Friend Sorting algorithm and MMBPSS, while "MMBPDSS" is better than DDS and saves almost 50% of the classical Selection Sorting. It really reaches the optimization purpose.

4.2 Comparison of Total Comparison Frequency

Table.2 shows the comparison of the MMBPSS, DSS and MMBPDSS algorithms with the classic Selection Sort algorithm and with the new Friend Sorting algorithm with respect to average of comparison numbers each algorithm takes to perform sorting.

Table 2. Total Comparison Frequency.

Array size	Average of comparison numbers (Test on different array of size)				
	Selection sort	New friend sorting algorithm	Min-Max Bidirectional Parallel Selection Sort	Dynamic Selection Sort	Min-Max Bidirectional Parallel Dynamic Selection Sort
1000	499500	500500	10871	415369	621162
3000	4498500	4501500	35597	3683523	3929732
5000	12497500	12502500	69541	10096142	9745052
7000	24496500	24503500	97718	19993926	21634554
10000	49995000	50005000	151299.3	40832313	39379276
30000	449985000	450015000	490405	368623040	303589600
50000	1249975000	1250025000	880382	1031800452	1018170886
70000	2449965000	2450035000	1319601	2032017833	2256552865
100000	4999950000	5000050000	1926387	4186028005	4102486376

Graphical view for Table.2 is presented in Figure. 12.

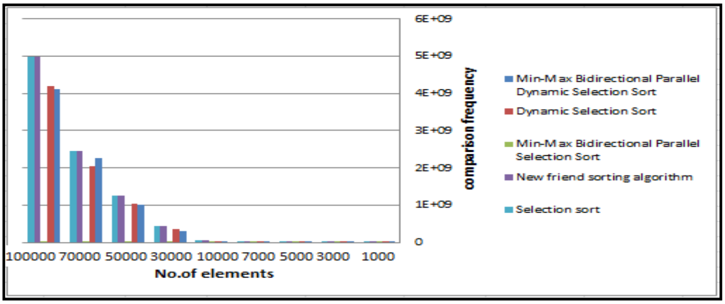


Fig. 12. Total Comparison Frequency.

It can be observed from the above graph that the total comparison frequency of Selection Sort and the new Friend Sorting algorithms are the same, while DSS and MMBPDSS reduce the total comparison frequency but MMBPSS perform the least number of comparisons in sorting procedure.

4.3 Comparison of Total Swapping Frequency

Table.3 shows the comparison of the MMBPSS, DSS and MMBPDSS algorithms with the classic Selection Sort and the new Friend Sorting algorithms with respect to average of swapping frequency each algorithm takes to perform sorting.

Table 3. Total Swapping Frequency

Array size	Average of swapping numbers (Test on different array of size)				
	Selection sort	New friend sorting algorithm	Min-Max Bidirectional Parallel Selection Sort	Dynamic Selection Sort	Min-Max Bidirectional Parallel Dynamic Selection Sort
1000	999	1000	994	999	1003
3000	2999	3000	2993	2999	3000
5000	4999	5000	4991	4999	5002
7000	6999	7000	6991	6999	6999
10000	9999	10000	9988	9999	10002
30000	29999	30000	29990	29999	30000
50000	49999	50000	49989	49999	50002
70000	69999	70000	69987	69999	70000
100000	99999	100000	99980	99999	100000

Graphical view for Table.3 is presented in Figure. 13.

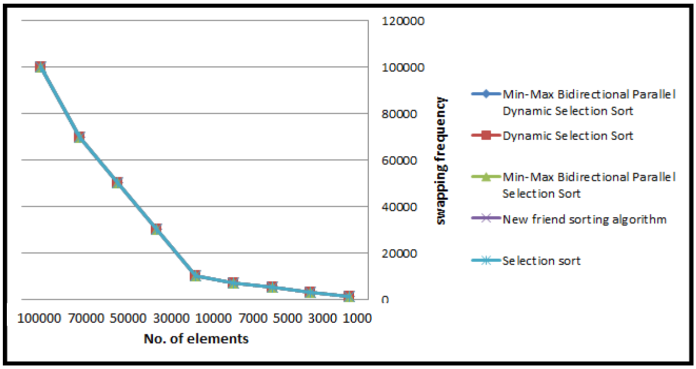


Fig. 13. Total Comparison Frequency.

It can be observed from the Figure. 13 that the classic Selection Sort, the new Friend Sorting Algorithms, MMBPSS, DSS, MMBPDSS perform the same number of swaps as the number of elements to perform sorting.

5. CONCLUSION

In this study, we present three new sorting techniques: "MMBSS","DSS" and "MMBPDSS" for selection sort that are tested and analyzed against the classical Selection Sorting and the new Friend Sorting techniques[3] to provide their efficiency. The graphs show that "MMBPDSS" save almost 50% of the classical Selection Sorting with 100% accuracy of order which get the benefit from effective utilization of CPU by using parallel computing with cost of increasing amount of space.

References

[1] **Min, W.,** "Design and analysis on bidirectional selection sort algorithm," in *Education Technology and Computer (ICETC), 2nd International Conference* on, Vol. **4**, pp: V4–380, (2010).

[2] **Bailey, D. A.,** Java Structure: *Data Structure in Java for Principled Programmer*, 2^{nd} ed. McGraw-Hill, (2003).

[3] **Iqbal, S. Z., Gull, H.** and **Muzaffar, A. W.,** A new friends sort algorithm. *In Computer Science and Information Technology, 2nd IEEE International Conference on*. pp: 326-329, ICCSIT, (2009).

[4] **Lakra, S.** and **Divy**, "Improving the performance of selection sort using a modified double-ended selection sorting", *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Volume 2, Issue 5, and May (2013).

[5] **Agarwal, A., Pardesi, V.** and **Agarwal**, **N.,** " A New Approach To Sorting: Min-Max Sorting Algorithm", *International Journal of Engineering Research & Technology (IJERT)* Vol. **2** Issue 5, May (2013).

[6] **Donald, E. K.,** The art of computer programming, *Sorting and searching*, **3**, 426-458.

[7] **Lipschutz**, **S.,** Theory and Problems of Data Structures, Schaum's Outline Series: *International Edition*, McGraw (1999).

تحسين خوارزمية ترتيب "الاختيار" باستخدام مفهومي الحوسبة المتوازية والبرمجة الديناميكية

خالد ثابت، وأفنان باوزير

قسم علوم الحاسبات، كلية الحاسبات وتقنية المعلومات

جامعة الملك عبدالعزيز، جدة، المملكة العربية السعودية

afnan-bawazir@hotmail.com

المستخلص. قد أجريت العديد من الأعمال البحثية لاكتشاف أفضل تحسين لخوارزمية ترتيب "الاختيار"، مثل خوارزمية ترتيب الاختيار الثنائية الاتجاه كخوارزمية "ترتيب الاختيار الصديقة" والتي يمكن وضع عنصرين في كل جولة، لقد قمنا بتحسين هذه الخوارزمية باستخدام مفهوم الحوسبة المتوازية، هذه الخوارزمية تسمى أصغر–أكبر ثنائي الاتجاه المتوازية للترتيب بواسطة الاختيار (MMBPSS). كما تقترح هذه الورقة استخدام البرمجة الديناميكية (المكدس) لتقليل وقت الفرز عن طريق زيادة مقدار مساحة الذاكرة. الفكرة الأساسية وراء استخدام المكدس هو القضاء على التكرار الذي لا داعي له في البحث عن العنصر الكبير والصغير. هذه الخوارزمية تسمى ترتيب الاختيار المتغيرة (DSS) ولدمج مزايا (DSS) مع مزايا "MMBPSS"، اقترحنا خوارزمية جديدة ثالثة تسمى أصغر–أكبر ثنائي الاتجاه المتوازية المتغيرة للترتيب بواسطة الاختيار "MMBPDSS". والتي تمكن من وضع عنصرين من عناصر الحد الأدنى والحد الأقصى من اتجاهين باستخدام خوارزمية الاختيار المتغيره في كل جولة بالتوازي، وبالتالي تقليل عدد الجولات المطلوبة للترتيب. وتم تقديم النتائج التي تم الحصول عليها بعد التنفيذ على شكل رسوم بيانية مع الهدف لاظهار "MMBPDSS" هي أفضل بـ٥٠٪ من خوارزمية ترتيب الاختيار العادية.

JKAU: Comp. IT, **Vol. 2**, pp: 45 - 56 (2013 A.D./ 1435 A.H.)

DOI: 10.4197 / Comp. 2-3

Guaranteed QoS Routing Scheme in MPLS -Wireless Access Networks

Afaf Mustafa Kotti

Jeddah University, Alkamil Branch, Alkamil, Saudi Arabia

akotti@uj.edu.sa

Abstract. With the increasing deployment of mobile network infrastructure and the emergency of real time applications, using Multi Protocol Label Switching (MPLS) capabilities is becoming more and more necessary to satisfy end user requirements in term of Quality of Services. In this paper, we propose a new algorithm that routes efficiently Label Switched Paths (LSP) in mobile MPLS network. The particularity of our work consists of integration both macro and micro mobility management in order to improve the overall performance of the system. We propose first a fast handoff process, which reduces service disruption. We present then new architecture which looks at avoiding triangle routing problem that MPLS inherits from Mobile IP. Finally, we integrate route optimization process which computes efficiently the shortest path by compromising between several Traffic Engineering (TE) objectives, like bandwidth guarantees, load balancing, and minimizing path hop count.

Keywords: MPLS, Mobile networks, handoff, QoS, traffic engineering, routing LSPs.

1. Introduction

As the world is becoming more dependent on wireless and mobile services, the question of the network's ability to handle such growing demand is getting more attention. The Internet Engineering Task Force (IETF) proposes Mobile Internet Protocol (IP) as the principal mechanism in IP-based wireless networks. However, Mobile IP presents several inconveniencies such as large signaling load for frequent registration updates and long handoff latency. Multi-Protocol Label Switching (MPLS) has been developed to overcome the limitation of conventional routing protocols.

MPLS allows the specification of explicit routes through the network, so-called Label Switched Paths (LSPs). Other advantage of MPLS includes service differentiation and traffic engineering. The distinguished advantages of MPLS have inhaled some researches in the literature on the use of this technology in the wireless infrastructure.

[1-3] proposed a mobile MPLS architecture which can be efficient but the proposed scheme is not suitable for intra-domain mobility, called micro-mobility.

To overcome this limitation, a mobile MPLS framework has been proposed by [4] called hierarchical MPLS. This scheme is too complicated and difficult to be applied in real time mobile transmissions.

We propose in this paper a new mobile MPLS architecture that supports both inter and intra Label Edge Router (LER) handoff. Our proposal also takes the advantages to improve MPLS routing paths. In fact, we have first proposed a new scheme able to avoid triangle routing, a problem that MPLS inherits from IP.

Then we have ameliorate our solution by introducing a new computing path algorithm that compromise between several traffic engineering objectives.

The rest of the paper is organized as follows. Section 2 introduces the proposed registration process in Mobile MPLS network. In section 3, we illustrate the two variants of Handoff support: inter LER, and intra LER

Handoff. Section 4 describes the proposed framework that essentially looks at route optimization process.

In this context, our proposal focuses first on how to eliminate triangular routing and then how to provide a sophisticated constrained routing algorithm for Real Time Internet applications. The final section contains our concluding remarks.

2. Registration Process in Mobile PLS Network

Mobile MPLS is based on the same standards that are already defined and applied in mobile IP. But using MPLS instead of IP will improve the Quality of Services (QoS) services which are required nowadays. Figure 1 presents a typical topology[5] for Mobile MPLS network. We use in next paragraph this topology to explain registration process.

When powering, the station Main Host (MH) searches the beacon signals from the surrounding Base Stations (BSs), and selects the one with the strongest signal as its serving BS. The mobile MH knows whether the selected BS is in its home domain or not on the basis on BS's address.

The MH sends a mobile IP registration to the nearest Label Edge Router/Foreign Agent (LER/FA). The latter notes the MH home address in its routing table and then transmits the registration message to the Label Edge Router Gateway (LERG) of its domain. The LER/FA label table (called Label Forwarding Information Base or LFIB) is illustrated in Table 1.

When the LERG gets the registration message, knows the IP address of current LER/FA, it sends the registration message to the Home Agent (HA) of the MH. The LERG uses its IP address to perform the global registration for inter domain mobility. Then, the LERG establishes an LSP between it and the current LER/FA with address LER/FA as Forwarding Equivalent Class (FEC). The LERG Label table is illustrated in Table 2.

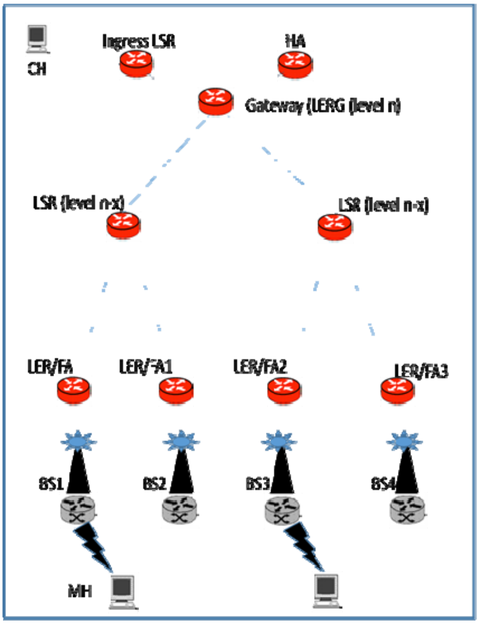


Fig. 1. Network topology.

The Home Agent makes an entry for the LERG in its Label Forwarding Information Base (LFIB), the entry for the LERG is shown in Table III which illustrates the HA label forwarding table. The latter has also an entry for the CH's label switch router (called Ingress LSR in Figure 1). Here Label Distribution Protocol (LDP) can be used to establish the downlink and uplink LSP.

Finally the LERG transmits the Registration reply message, sent from the Ingress LSR from the HA to the MH along the established LSP. The resulting Label table of the LERG after registration is shown in Table IV.

3. Handover Support in Mobile MPLS Network

In the literature, the handover mechanism is classified in Mobile MPLS access network into two types. intra-LER handover and inter-LER handover[2].

When the mobile station is moving, it passes between two BSs. If these BSs are under the same LER/FA, the handover is called inter-LER handover. However, it is called intra-LER in case of the new BS and the old BS are managed by different LER/FAs[6]. We present in the next paragraph how to manage handover requests.

3.1 Intra LER Handover

During a call, the mobile station MH moves between different cellular systems which are controlled by different base stations. The MH transmits a message to the current LER/FA informing through it its movement. The LER/FA, upon receiving the movement signaling message, begins buffering in-flight packets.On the other hand, the mobile station, searching for a new BS, finds one and registers at layer 2 with that BS.

Now, the mobile should verify if it is still under the same network. For this reason, when receiving an advertisement message from the LER/FA it takes advantage of the LER/FA IP address. If it is still under the same subnet, the LER/FA stops buffering and transmit in-flight packets designated to MH toward the new BS. It is important to mention here that no change occurs to LERG Label table after an intra-LER handover since no message has been sent to the LERG. Figure 2 illustrates intra LER handoff in Mobile MPLS network.

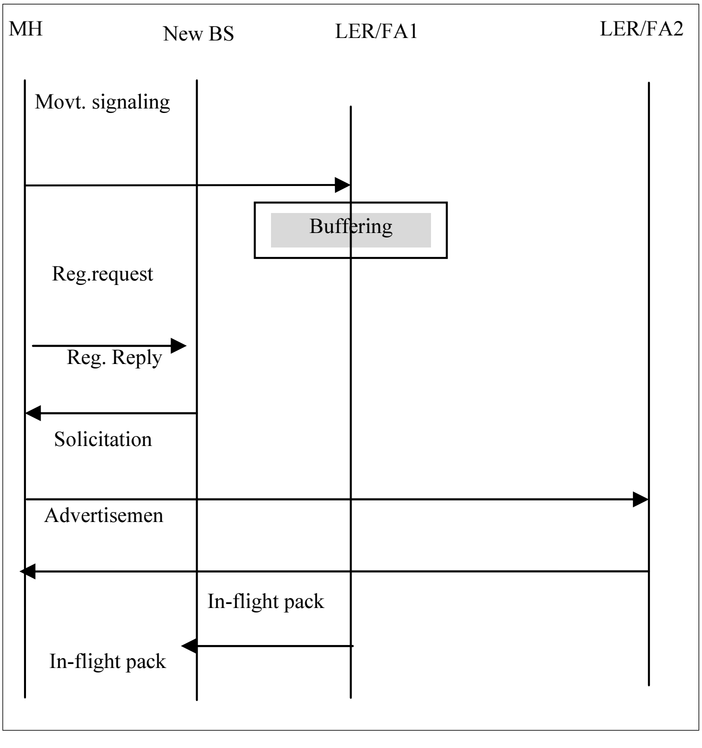


Fig. 2. Intra LER handoff.

3.2 Inter LER Handoff

When the MH moves to a new IP subnet, it first registers to the new Label edge router (called new LER/FA) by sending a registration request message. Then, it conducts the same processes seen in section 2 for registration.

The mobile station should now recuperate all packets that are destined to it and already reach the old LER/FA. For this, it sends to the old LER/FA a message through the new LER/FA informing it by the handover.

When receiving this message, the old LER/FA stops buffering and transmits in-flight packet to the mobile station. Figure 3 illustrates Inter LER handoff in Mobile MPLS.

We illustrate the LERG Label Table in Table 5.

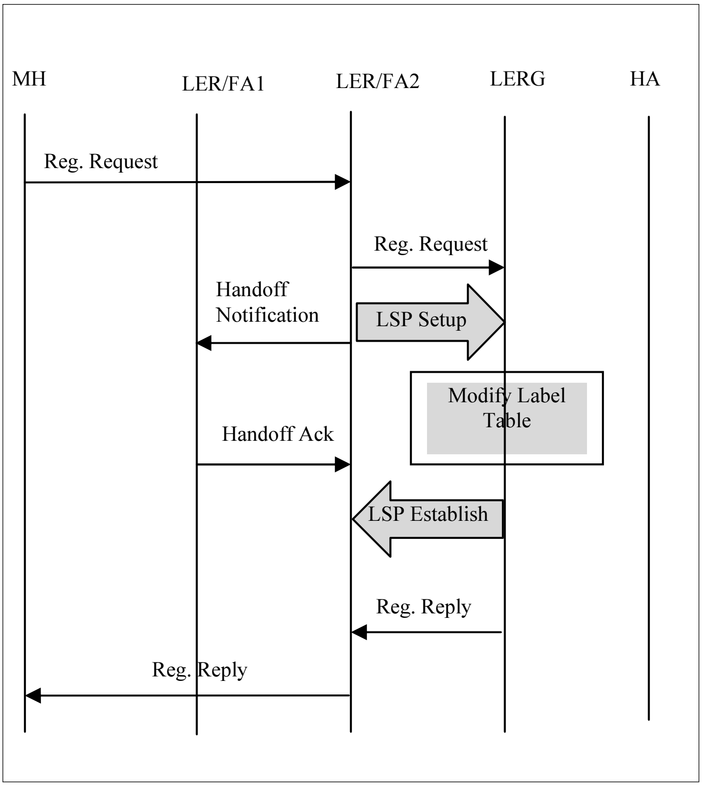


Fig. 3. Inter LER handoff in Mobile MPLS.

Table 1. Label table (LFIB) of LER/FA.

In Port	In Label	Fec	Out Port	Out Label
0	_	@MH	1	1

Table 2. Label table (LFIB) of LERG.

In Port	In Label	Fec	Out Port	Out Label
1	1	@LER/FA	3	3

Table 3. Label table (LFIB) of HA.

In Port	In Label	Fec	Out Port	Out Label
1	3	@LERG	3	9
2	10	@Ingress LSR	2	11

Table 4. Label table (LFIB) of LERG.

In Port	In Label	Fec	Out Port	Out Label
1	1	@LER/FA	3	8
0	11	@MH	3	8

Table 5. Label table (LFIB) of LERG after inter LER handoff.

In Port	In	Fec	Out Port	Out Label
1	1	@LER/FA	3	8
2	14	@MH	1	15
2	14	@LER/FA2	1	15

4. Route Optimization Process

We are interested in this paper to improve the efficiency of the mobile MPLS network by introducing new mechanism in the routing process. We first discuss how triangle routing is established in mobile MPLS scheme and provide solution to avoid such a problem. Then we propose a new scheme to calculate efficiency the most optimal path to reach destination node.

Let's begin in the next paragraph by explaining our proposed solution for eliminating triangle routing.

4.1 How to Avoid Triangle Routing?

When a mobile station wants to communicate with another one, called here correspondent host (CH), it should obligatory register to its home agent. The latter will be responsible for forwarding data packets to CH following a specific Label Switched Path (LSP). In this way, a triangle route is formed between the source node, the home agent and the destination node. This can affect badly the resource management process and reduce the performance of the entire network. In fact, the data had to follow long path with a number of intermediary nodes to the HA, resulting into a significant delay.

Our proposal consists on the following scheme. When MH moves to a foreign network, it registers to the HA with its new temporary address. The HA keeps this address in its cache. Thereafter, HA will forward this update information to all edge routers in the network. The ingress routers store the binding update between the permanent IP address of the MH and its temporary address in its binding cache and starts a timer. When this timer runs out, that the ingress LSR removes the binding from its cache and send a request to the HA for a new update if cache is available. Thus, every time whenever the HA receives any update from the MH, it will forward this binding update to ingress LSRs so that the movement of MH is transparent to these agents and can provide optimized paths to intended destination.

Our second proposal is to involve QoS constraint based routing with Mobile MPLS architecture. Our goal consists basically of finding a feasible path from source (Ingress LSR) to destination (LER/FA) if one exists, and to select the one that achieves efficient resource utilization if more than one path is available. This will be the topic of the next paragraph.

4.2 Path Computation

Shortest Path First (SPF) routing protocol is the most commonly used protocol in IP/MPLS networks[7]. The core algorithm is based only on the number of hops to compute the shortest path between the ingress and egress router. In despite of its simplicity, SPF protocol presents several inconveniencies. In fact, it generally leads to an inefficient usage of the network infrastructure. It can create bottleneck in some links while other link are severely under-utilized.

Some researchers [4, 8-10] are interested in how to distribute the load in the network to avoid network congestion. However, their proposals selects paths with high number of hops. The trade-off between two or more than two traffic engineering objectives looks to be so difficult to realize. From this fact arises our motivation to compromise between network load balancing and path hop count.

Our problem is a multi-objective optimization problem. Several methods in the literature exist to solve such a problem, like genetic algorithm, but most of them are very complicated and difficult to apply in real time transmission networks. For this reason, we propose to use weighted criteria method. Here, we will form an objective function which will correspond to the traffic engineering metric of the link and we then calculate the shortest path on the basis of this function.

Before defining the objective function form, we need to define some notations. The network is modeled as a graph *G* which is composed of a set of nodes and links between the different nodes. We note *e* a link. Each link in the graph is characterized by a set of parameters: the source node, destination node, bandwidth capacity *C(e)* and residual bandwidth *R(e)*. We formulate in the following expression the proposed link metric *Tem(e)*.

$$Tem(e) = \alpha \frac{R(e)}{C(e)} + 1 \text{ where } \alpha \in [0\ 1] \qquad (1)$$

The first term ($Tem(e) = \alpha \frac{R(e)}{C(e)}$)in the expression above will be representative of load balancing objective. However, the second one is for reducing path hop count. The parameter α varies from 0 to 1 depends on the routing objectives of the network. For example, to give the same priority to both load balancing and path hop count, we suggest α to be equal to 1.

5. Pseudo Code of the Route Computation Algorithm

We present in this section the pseudo code of the route computation algorithm. The algorithm will return the path taken by an LSP from the source to the destination minimizing the path objective function. So, we use the well-known Dijkstra scheme and adapt its formulation to our need. Our route computation algorithm is detailed in Figure 4.

Using our proposal, traffic in MIP is conducted via optimal paths that are labeled using MPLS labeling mechanism. This will optimize considerably network resource utilization and guarantee more and nore quality of services.

1 **Route-computation-algorithm procedure(G, request(source, destination, bandwidh required))**
{
2 Calculate the objective function related to each link in the network.
3 Eliminate from the network all links that haven't enough residual bandwidth. Consider in the rest of the algorithm the resulting reduced topology with remaining links and nodes.
4 Use Dijkstra algorithm to compute the shortest path in the network using *Tem(l)* function as a link metric.
5 Route the request from source until destination along that path
}

Fig. 4. Pseudo Code of the Route Computation algorithm.

6. Conclusion

This paper addresses a framework of mobility management and traffic engineering for Mobile MPLS networks, called Global Mobile MPLS (GM-MPLS). The proposed scheme has the advantage to involve two ranges of network mobility, that are micro mobility and macro mobility, to provide excellent solution to the problem of mobility support in wireless environment.

We have presented registration procedure and discussed handoff techniques: inter LER and intra LER handoff. These techniques reduce considerably the number of lost packets due to movement of MH in mobile MPLS network. We have also proposed a new routing scheme to support traffic engineering. The algorithm satisfies meeting QoS requirement of bandwidth, efficient usage of network infrastructure, by providing network load balancing and reducing path length. Thus we can conclude that our solution improves the overall performance of the mobile system.

References

[1] **Ren, Z.**, **Tham, C.**, **Foo, C.** and **Ko, C.**, Integration of Mobile IP and multiprotocol label switching, *In IEEE international Conference on Communication (ICC)*, Finland, (2001).

[2] **Kohli**, **J.** and **Kumar Rai, M.,** The improved Route Optimized Mobile MPLS Technique with Hand-off, International *Journal of Applied Information System (IJAIS)*-ISSN: 2249-0868, Vol. **2**-No.5 (2012).

[3] **Sthom, K., Afifi H.** and **Pujolle, G.,** Wireless MPLS: a new layer micro mobility scheme, *In ACM MobiWac 2004*, Philadelphia, PA, (2004).

[4] **Yuan**, **X., Kang, L.** and **Chen, Y.,** Mobile IP Network Based on Hierarchical MPLS, *International Conference on Wireless Communications, Networking and Mobile Computing,* **2**: 104-7, (2006).

[5] **Chiussi, F.M., Khotimsky, D.A.** and **Krishnan, S.,** A network architecture for MPLS-based micro mobility with MPLS, *In IEEE WCNC02*, Orlando, FL, (2002) .

[6] **Chumchu**, **P., Sirisaingkarn, S.** and **Mayteevarunyou, T.,** Performance Analysis and Improvement of Mobile MPLS, *International Conference on Information Networking (ICOIN)*, pp: 317-22, (2011).

[7] **Guerin**, **R., Orda, A.** and **Williams, D.,** QOS routing mechanisms and OSPF extensions, *IEEE GLOBECOM'97*, **3**: 1903-1908, (1997).

[8] **Kotti**, **A., Hamza, R.** and **Bouleimen, K.,** Bandwidth Constrained Routing Algorithm for MPLS Traffic Engineering, *International Conference on Networking and Services - ICNS* , p. 20, (2007).

[9] **Mishra, P.P.** and **Saran, H.,** Capacity management and routing policies for voice over IP traffic, *IEEE Network*, **14**: 20-27, (2000).

[10] **Savinya Polvichai** and **Prawit Chumchu,** Mobile MPLS with route optimization: The proposed protocol and simulation study, *Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp: 34-39, (2011).

مخطط لضمان جودة خدمة توجيه الرسائل في شبكات MPLS اللاسلكية

عفاف مصطفى القطي

جامعة جدة، فرع الكامل، الكامل، المملكة العربية السعودية

akotti@uj.edu.sa

المستخلص. تزامنًا مع التطور الكبير الذي تشهده البنية التحتية لشبكة الاتصالات اللاسلكية وانتشار التطبيقات المقيدة بالزمن، استخدام البروتوكول (MPLS) أصبح أكثر فأكثر ضرورة نظرًا لما يجلبه من مزايا متعدد لتلبية متطلبات المستخدم النهائي من حيث جودة الخدمات. في هذا البحث، نقترح خوارزمية جديدة لإنشاء المسارات (LSP) داخل شبكة MPLS النقالة. خصوصية عملنا تكمن في إدارة التنقل الكلي والجزئي على حد سواء من أجل تحسين الأداء العام للنظام. نقترح أولا تقنية تمكن المستخدم من التحول بسلاسة وبسرعة، مما يقلل من انقطاع الخدمة. نقدم في مرحلة ثانية طريقة جديدة لتجنب مشكلة مثلث التوجيه التي يرثها MPLS من بروتوكول الإنترنت IP. وأخيرًا يعرض البحث اقتراح لتحسين المسارات، وذلك بالأخذ بعين الاعتبار جوانب متعددة مثل ضمانات عرض النطاق الترددي، موازنة الحمل بين المسارات وتقليل عدد القفزات بالمسار.

JKAU: Comp. IT., **Vol. 2**, pp: 57 - 82 (2013 A.D./ 1435 A.H.)

DOI: 10.4197 / Comp. 2-4

SCADM: A Developed Smart Classroom Acoustics Design Model for Enhancing Educational Environment

Abdul Hamid M. Ragab, Amin Y. Noaman, Ayman I. Madbouly

Faculty of Computing and Information Technology,
King Abdulaziz University,
P.O.Box. 80221, Jeddah 21589,
Kingdom of Saudi Arabia
aragab@kau.edu.sa

Abstract. The quality and efficiency of learning and teaching processes in academic universities depend on many important factors. Acoustic quality within the learning environment of the classroom and its surroundings is one of the most important factors that affect the quality of the learning process. Several studies and researches focused on the subject of acoustical comfort in university classrooms. In this paper we study smart classroom acoustics design issues required to achieve high quality acoustics conditions. These designing issues include inside and outside noise sources, designing for optimum reverberation time, selection of sound insulation and acoustical treatment materials, designing for speech intelligibility, and auralization inside the KAU classrooms. The aim is to suggest a developed smart classroom acoustics design model (SCADM) that can be used by architects, acoustics engineers and designers in an early stage of classroom design in order to achieve the acoustical conditions of KAU university classrooms. The goal of this research is to raise the quality and efficiency of the educational environment to reach an excellent learning environment, and hence increasing students learning outcomes.

Keywords: Smart classroom, education quality, Students learning outcomes, acoustical comfort in university classrooms.

1. Introduction

Formal education is conducted in classrooms, in which there is an intensive amount of verbal communication between students and teachers and among students themselves during this learning process. The existence of high levels noise in the classroom will affect the learning and teaching environment for both students and teachers, and will make students tired prematurely. This may result in consuming students' cognitive abilities that could be better employed to understand and pay more attention to the content of their classes. Building classroom with good acoustics is one of the important considerations while designing new classrooms. Decades of research have been devoted by architects, audiologists, acousticians, and speech language pathologists to document the educational value of acoustics quality and the detrimental effect of bad acoustics design on students' learning and achievements.

Smart classrooms are technology based enhanced classrooms. Smart classrooms provide the opportunity to integrate new learning technologies to enhance the learning process for both teachers and students. Applying new technologies in learning help students to actively learn and inspire them to use technologies for learning throughout their lives. Smart classrooms present the learning environment with equipment such as interactive smart board, data projector, audio amplification system, media and video, smart tables, computers, document camera, and VCR/DVD players. Existence of such devices inside smart classroom increases the overall background noise (BN) by 6 to 10 dBA[1].

Many research studies have been carried out to highlight the importance of having good acoustic design conditions to achieve good sound intelligibility inside classrooms. Acoustical measures of classrooms, such as speech intelligibility[2], background noise levels[3,4], early decay times[5], reverberation times[6], as well as various early/late sound ratios have been the focus of these studies.[7]

In this paper, interrelationships of these measures will be considered to evaluate the appropriate design model of smart classrooms. The suggested smart classroom acoustics design model can be used by architects, acoustics engineers and designers in an early stage of classroom design in order to achieve the optimal acoustical conditions of KAU university classrooms.

2. Literature Review

With the evolution of modern era, there were many aspects that affect the acoustic performance inside the classroom environment. In the past, the environment in the classrooms were pleasant, quiet, and enjoyable. Today, modern classrooms become more reverberant and have more noise. Such acoustic problems were existed because of the insufficient awareness and the shortage of perception of the problem on the part of the professionals involved. Although there is no lack of resources or funds, they had lack of solutions to such problems [1]. It showed that the best way to solve such problems is to avoid them in the design phase. Many researches proved negative impacts and effects of noise, the lack of clarity of talk, and the lack of speech intelligibility not only on the efficiency of learning and the quality of teaching but also on the well-being of students and teachers. Students are impaired by background noise and teachers suffer from raising their voice level to compensate for the high level of background noise and increase the signal to noise ratio (SNR) [8, 9]. On the other hand, if the classroom acoustics were well designed and the acoustical properties inside the classroom were improved, this will result in an improvement on learning and students' behavior, and these results are registered in numerous studies [10, 11].

Wróblewska *et al.*[12] studied the influence of acoustical adaptation on classroom's acoustical environment; they showed that acoustical adaptation in the classroom changed the values of acoustical parameters into the desired range. They showed that not only the amount of absorption plays a role, but also its placement is essential.

Dockrell and Shield[13], studied the effects of using sound field systems (SFS) in classrooms. They found that small and subtle effects could be noticed when SFS is used. Their study showed that using SFS in classrooms improved the speech to noise ratio (SNR) and caused a gain in the spoken comprehension measure. However, further studies are required to substantiate this effect, especially in poorly acoustics classrooms. They suggested that further work and studies for classroom acoustical parameters are needed to be considered as an additional variable while measuring the benefits of using SFS in classrooms.

In a study done by Rand[14], for the Heating, Ventilation, and Air Conditioning (HVAC) systems and classroom acoustics, they showed that installation cost will not greatly increase if the acoustical standard is met by using standard HVAC equipment. They also mentioned that to meet the requirements, we need to do good design, selection and application practice as follow[14]:

- Make sure that accurate sound power level (SPL) is obtained. This will help to reduce the need of a large factor of safety in the design.
- Predict classroom sound levels and make sure that adequate attenuation is achieved by performing the required acoustical analysis.
- Find good places to locate the equipment away from the classroom. A good place is in or above less critical places such as utility areas, corridors, or mechanical rooms.
- Appraise the trade-off between performing path attenuation and buying quieter equipment.
- Minimize regenerated noise by lowering the airflow velocity in the ductwork.
- Make sure that you follow the manufacturer's design recommendations while designing and installing diffuser.

The subject of acoustical comfort (ambient noise, sound insulation, reverberation time, speech intelligibility, auralization, acoustical materials) in classrooms of primary and secondary schools, as well as in university classrooms has been the focus of several studies around the world [15–18].

In all studies, the acoustic quality of the classrooms were analyzed based on measurements of the sound pressure level inside and outside the classrooms, the reverberation time, and sound insulation inside the classroom.

Zannin and Zwirtes[17] evaluated the acoustic performance of classrooms in public schools. The acoustics quality of classrooms of three constructive designs has been evaluated. They analyzed the quality based on measurements of the reverberation time, sound pressure level inside and outside the classrooms, and sound insulation. Results revealed design errors in all the schools of this study which caused poor acoustical quality of the surveyed classrooms, for all three constructive designs studied. These errors involved both the architectural design and the materials used in the interior finish of the schools. They showed that the surveyed designs do not meet the guidelines of either the Brazilian Standards or the International Standards employed as references.

Measurement of reverberation time and sound insulation usually follows the international Standards ISO 140-4, ISO 140-5, ISO 717-1, and ISO 3382. Also, the American National Standards Institute (ANSI) approved ANSI Standard S12.60 for Classroom Acoustics [18], titled "Acoustical Performance Criteria, Design Requirements, and Guidelines for Schools," is widely used in such cases. This standard presents an enhanced learning environment for students and teachers alike by improving the classroom acoustics design conditions for good speech intelligibility. In our research, results (sound insulation and reverberation time) will be compared with reference values found in the ANSI Standards S12.60.

3. Noise Effect of Smart Classroom

In all researches, studying and analyzing the acoustic quality of the conventional classrooms depends on measuring the sound pressure levels inside and outside the classroom. Also, it depends on measuring the reverberation time, and sound insulation used inside the classroom. Usually, the international Standards ISO 140-4, ISO 140-5, ISO 717-1, and ISO 3382, are commonly used while measuring the reverberation time and sound insulation inside classrooms. Also, the American National Standards Institute (ANSI) has approved ANSI Standard S12.60-2002 for acoustical features of learning places and design requirements for classroom acoustics.

Thomas[19] has defined smart classroom as "an interactive multimedia electronic classroom networked to the Internet and housing a video/audio, and broadcast on demand system". Smart classrooms equipped with interconnected computers for each student, allow for interactive and collaborative learning. Smart classrooms present collaborative and interactive learning environment equipped by new technology devices and equipment such as interactive electronic smart board, instructor's lectern equipped with data projector, audio amplification system, media and video projector, smart tables, networked computers, document camera, VCR/DVD player. The existence of such additional components inside smart classrooms makes them different from conventional classrooms. The existence of such equipment increases the noise level inside smart classroom and this effect must be taken into consideration while designing a new classroom or make acoustical treatment for an existing classroom.

In the proposed model; explained in section V; different layouts of KAU smart classrooms will be investigated. Since, ANSI standard (ANSI S12.60-2002) doesn't include the effect of smart classroom instructional equipment, it is important to study this to see how this will affect the learning process.

4. Importance of the Study

Speech intelligibility usually degrades with the existence of excessive noise and reverberation inside classrooms. This degradation of speech intelligibility will result in reducing students' understanding and hence reduces students' learning and teaching quality. Speech intelligibility tests usually measure the percentage of words that a listeners with normal hearing can correctly understand from a list of spoken words. A speech intelligibility rating of 75% means that listeners with normal hearing could understand only 75% of the words read from a list [2, 20].

Although the need of good acoustically designed classrooms and the way to achieve this have been known for decades, unfortunately this information was not available for architects and planners engineers, as well as for administrators and decision makers. While solutions are not

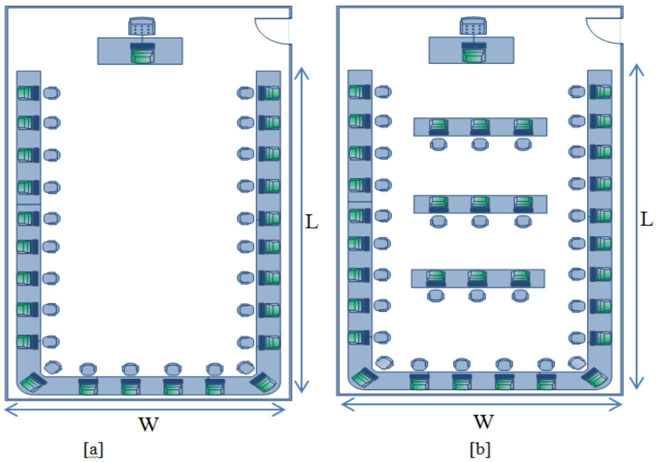


Fig. 1. (a, b): Examples of Different Smart Classroom Layouts.

prohibitively expensive the classroom acoustics problems seem to be endemic. Actually, the reason was the shortage of understanding of the problem and its solutions not the insufficient funds. The cost spent on school buildings nationwide was \$ 7.9 billion[20]. Good acoustic performance and listening conditions could be achieved with only a fraction of this amount if we take classroom acoustics design into considerations.

To achieve this, architects and planners should consider the classroom acoustics from the beginning of the design process. The optimum solution of many acoustics problems could be prevented beforehand not after it happened. Acoustics problems can be avoided during the design phase with low cost and by using different design arrangements of the same building materials. This is much better than renovation of a poorly designed classroom which may be more expensive.

Our research aimed to study and propose a developed classroom acoustics design software model of the acoustics characteristics of KAU existing classrooms and to suggest design recommendations to optimize the classroom acoustics of new KAU classrooms. The suggested model can be used by architects, acoustics engineers and designers in an early stage of classroom designs to evaluate the acoustical conditions of university classrooms. The study is suggested after complaints relating to the acoustics properties of the existing classrooms, since building classroom with good acoustics has to be one of the important design considerations for new classrooms.

5. Proposed Design Model (SCADM)

In this paper we propose different classroom's layout models that could be adequate for smart classrooms. In our model we considered the number of students in the classroom as a design factor. As shown in the proposed models below in Figure.1 (a, b) the number of students inside the classroom depends on the classroom dimensions. Researches [21, 22] showed that the lower middle range for social distance as a mean for calculating minimum size of the classroom is 7 feet (≈ 2.13 m). However, for smart classrooms this distance is large. Practically, we found that a distance of (1.4 m) is adequate and gives reasonable classroom dimensions for large and small groups of students. As shown in Figure 1a, the dimensions of the classroom can be calculated by the following equations that relate classroom dimensions with the number of students and hence the number of computers inside the classroom:

$$L\ (Length) = D * [N/G - 1 + R]$$

$$W\ (Width) = D * [N/G + 1]$$

where:

N≡ Number of students

G≡ Number of groups of computers

D≡ Recommended Social Distance (≈ 1.4 m)

R≡ Difference between the numbers of computers in length and width.

Figure.1 (b) shows a different classroom layout with the same dimensions but with an increased number of students by (N/G -1)*3. Of course, different classroom layouts could be arranged. However, there are other factors that may affect the maximum number of students inside classroom. First of these factors is the room length, since speaker's visual gestures can greatly improve the recognition of what he/she is saying. Benefit of this is not easily quantifiable but generally it is thought that the maximum distance from instructor to student should not be more than 20 m [23]. This is the maximum distance at which student can identify these visual prompts, and to also ensure that a good amount of direct sound level could be received by students. Second factor is the room volume; the ideal room volume per seat for classroom is "volume / seat 2 – 5 m^3 ".

In our previous research [24] we have developed a classroom acoustic assessment model in which we found that noise sources inside the classroom ranked with 23% along other criteria that affect the learning quality. Smart classroom equipment was also identified as common source of noise generated within new classrooms with an importance level of 20%. And this represents a ratio of 4.6% weight related to the whole model of criteria that affect the learning process and its quality. These results showed that how noise emitted from smart classroom equipment could affect the enhancement of learning quality and how this noise source could reduce the speech intelligibility inside smart classrooms.

Adding more computers means adding more noise sources. It's known that by adding or combining levels of 'n' equal loud sound sources the total sound level will increase. The total level in dB is the level of one sound source plus the increase of sound level in dB[25], as indicated in table 1. The total sound level is calculated using the following equation:

$$L_{\Sigma} = \Delta L + L_1 = 10 * log_{10}(n) + L_1$$

where,

L_{Σ}: total sound level
ΔL: Level increase ΔL for 'n' equal loud sound sources
n: number of equal loud sound sources
L1: level of one sound source

Figure 2 shows the relation between the number of equal loud incoherent sound sources *"n"* and the sound level difference Δ L. As shown in Figure 2, the sound level increases as the number of sources increases.

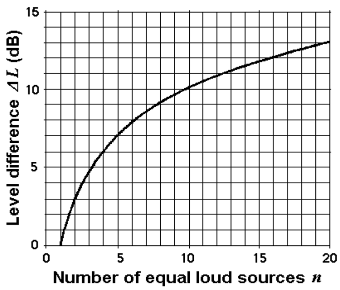


Fig. 2. Relation between number of equal loud sources 'n' and the sound level difference [19]

Table 1. Level increase ΔL for 'n' equal loud sound sources [25].

Level increase Δ L for 'n' equal loud sound sources	
Number of 'n' equal loud sound sources	**Level increase Δ L in dB**
1	0
2	3.0
3	4.8
4	6.0
5	7.0
6	7.8
7	8.5
8	9.0
9	9.5
10	10.0
12	10.8
16	12.0
20	13.0

6. Research Methodology

Our research investigations will be conducted in listening environments at KAU smart classroom models to identify good and poor classroom models. This will be investigated in details using the following methods:

- Speech perception tests in live and simulated background noise in classrooms
- Classroom noise levels daylong recordings [A-weighted Sound Level].
- Acoustical measurements of classroom acoustic parameters [Clarity, Signal to Noise Ratio (SNR), Reverberation Time (RT)].
- Applying some acoustical modifications/treatments to the classrooms that were identified by teachers as being poor acoustically (The addition of acoustic ceiling tiles and repeating the detailed measurements).
- Proposing a developed smart classroom acoustics design model that can be used by architects, acoustics engineers and designers in an early stage of classroom designs.

7. Classroom Acoustic Design Considerations and Standards

To design a classroom with optimum acoustical characteristics for a learning process, it is essential that both architectural and mechanical design should comply with the existing design standards and many important factors and considerations must be achieved. With the existence of excessive reverberation time or background noise in classrooms causing interference with speech communication and thus an acoustical impediment to learning will be present. In this section we present the acceptable standard level of reverberation time and the A-weighted and C-weighted sound level inside classrooms, see Table 2. Also, some considerations and guidelines that are intended to assist in achieving conformance to the reverberation time criteria and an acceptable noise level inside classrooms will be introduced.

Table 2. Limits on A- and C-weighted sound levels of background noise and reverberation time.

Learning space	Greatest one-hour average A- and C-weighted sound level of exterior source background noise (dB)	Greatest one-hour average A- and C-weighted sound level of interior source background noise (dB)	Maximum permitted reverberation times for sound pressure levels in octave bands with mid-band frequencies of 500, 1000, and 2000 Hz (s)
Core learning space with enclosed volume ≤ 283 m^3 (≤ 10 000 ft^3)	**35 / 55**	**35 / 55**	**0.6 s**
Core learning space with enclosed volume > 283 m^3 and ≤ 566 m^3 (> 10 000 ft^3 and ≤ 20 000 ft^3)	**35 / 55**	**35 / 55**	**0.7 s**
Core learning spaces with enclosed volumes > 566 m^3 (> 20 000 ft^3) and all ancillary learning spaces	**40 / 60**	**40 / 60**	**No requirement**

7.1 Classroom Reverberation Time

Reverberation time (RT60) is defined as the time required in seconds for sound to decay 60 decibels from its initial level after a source stops generating sound. It was used to define the optimum RT for an auditorium or room, which depends upon its intended use. According to ANSI standard, classrooms should have reverberation times in the range of (0.4-0.8) seconds, but many existing classrooms have reverberation times of 1 second or more. Figure 3 shows the suitable reverberation time for various room usages. As shown in Figure 3, classrooms need a

shorter reverberation time for speech to be understood more clearly. If the reflected sound from one syllable is still heard when the next syllable is spoken, it may be difficult to understand what was said [26].
[27]has developed an empirical reverberation time equation that relates the value of the reverberation time in a room to the room's volume, and the total sound absorption inside that room, as follows:

$$RT_{60} = \frac{4\, ln\, 10^6}{C} \frac{V}{Sa}$$

where,
C: the speed of sound in the room
V: the room's volume in m^3
S: the total surface area of room in m^2
a: is the average absorption coefficient of room surfaces
Sa: total absorption in sabins.

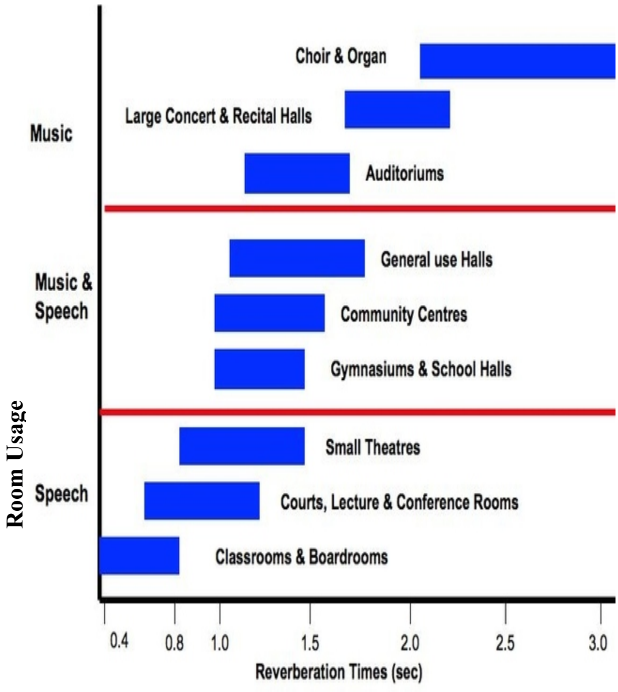


Fig. 3. Suitable RT (in seconds) for various rooms typically found in educational facilities [26].

It must be noted that:

- Total absorption and reverberation time can be changed based on the frequency defined by the acoustic properties of the space.
- The shape of the room and sound losses are not taken into account in equation[27]. It must be noted that this is important for large spaces.
- In lower frequency ranges the sound energy absorbed will be less, causing longer reverberation times at these lower frequencies.

In learning spaces, we can control the reverberation time value by using sound absorption materials. Selecting the place to add such absorbing materials and the area of materials used for treatment are important factor in controlling the reverberation time and hence in achieving optimum acoustical characteristics in classrooms and learning spaces [28].

Understanding of spoken words will be affected and reduced if there is an excessive reverberation inside the lecture room. On the other side, if there are too much sound absorbing treatment materials inside the classroom beneficial early sound reflections will reduce speech intelligibility for distant listeners as a result of the rapid fall off of the

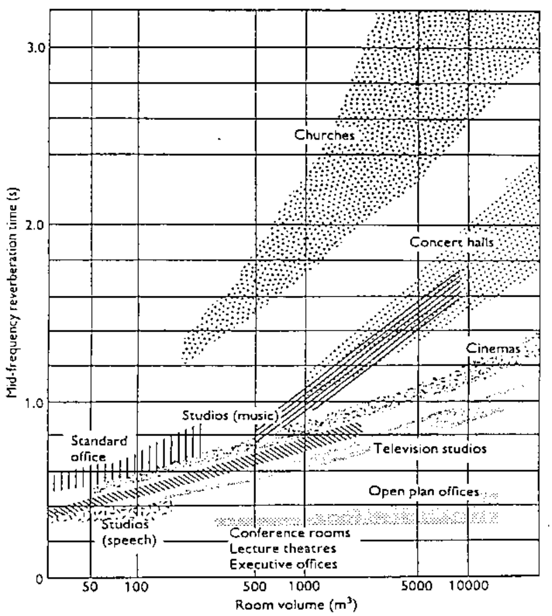


Fig. 4. Optimum RT –vs.- Room volume for different room usages [28].

speech levels from a talker with distance. Figure 4 shows the optimum reverberation time of mid frequency (500 Hz) in seconds of different room usages verses room volume.

7.2 Discrete Echoes

Reflected sound causes a special problem called discrete echoes. We are familiar with the echo phenomenon that is hearing one's voice answer a second late after shouting into a canyon. This kind of echoes can also happen in rooms, albeit more quickly [29]. If a teacher's voice is continuously echoing off the back wall of a classroom, the lecture will be difficult to understand because of each echo will interfere with the next word.

7.3 Early Reflections & Late Reflections

Sound reflections inside classrooms are important to increase speech intelligibility; however reflections arrived later are not desirable [29]. Sound reflections that continue to reflect and bounce inside classroom and arrive to the listener after (0.05) seconds of arriving of the direct sound cause one word to blur into the next word. This degrades the speech intelligibility inside the classroom. So, to design good acoustics classroom these later-arriving reflections must be acoustically treated, to increase speech intelligibility and to prevent additional sources of noise inside the classroom. Early arriving reflections are added to the direct sound while arriving to our hearing system. These early-arriving sound reflections are considered useful because they make the voice louder and hence increase speech intelligibility [30]. So, to achieve optimum speech intelligibility, classrooms must be acoustically designed to increase the amount of direct and early-arriving reflections of the speech sound to the listener. At the same time, the classrooms must be acoustically designed to decrease the amount of later-arriving speech sound reflections and decrease the amount of ambient noise as possible.

The ratio of direct and early arriving speech energy to later-arriving speech energy and ambient noise has been called a useful-to-detrimental sound ratio. These ratios are related to speech intelligibility scores [30], as well as to other more complex measures such as the Speech Transmission Index (STI) and the simplified version of it, Rapid Speech Transmission Index (RASTI) [31].

7.4 Flutter Echo

Another type of echo could happen when two flat, hard surfaces are parallel or between floor and ceiling. In these cases the sound rapidly bounces back and forth between these parallel surfaces and creates a ringing effect echo. This echo will interfere with hearing and this phenomenon is called flutter echo that can be dealt with as explained in [32].

7.5 Location of the Absorbing Material

In cases where there is no fixed lecture position for the teacher, we need to consider the following two cases for ceiling height:

- When ceiling heights ≤ 3 m (10 ft.) place most, if not all, of the sound-absorption treatment material on the ceiling. This is the best design option [33].
- When ceiling heights ≥ 3 m (10 ft.); in such case we need to place an increased amount of sound-absorption material on the walls. If almost all of sound-absorbing materials are installed on the ceiling, then it is prudent to use furnishings like bookshelves of sufficient height to ensure that the sound waves traveling across the room are spread and scattered in the direction of the sound-absorbing ceiling.

7.6 Mounting of Acoustical Treatment in Classrooms

The airspaces behind the wall or above the ceiling acoustic treatments are usually specified by the architect. These airspaces have a minimum value that is specified by the manufacturers of these acoustic treatment materials. As long as the minimum airspace required for installing treatment materials exist, the actual sound absorption coefficient of these material should not be less than the published values at frequencies of 500 Hz and higher [34]. Wall-mounted treatment materials should be installed using glue or clips to the wall surface as manufacturer recommends, or these can be fastened to added spacers to achieve sound absorption coefficients stated.

7.7 Carpeting in Classrooms

Using carpets in classrooms can help significantly to reduce background noise levels from chair and foot impacts, and can also reduce impact noise transmitted to room below [36]. What must be noticed is that carpets alone cannot provide the required sound absorption for classrooms especially at low frequencies since its absorption is generally poor at these frequencies.

7.8 Absorption of Furnishings and Occupants

When calculating classroom reverberation time, sound absorption coefficients of typical furnishings such as tables, desks, chairs, and storage cabinets must be included. Experiments comparing the reverberation for furnished and unfurnished classrooms showed that a sound absorption equal to 5% of the floor area is a conservative approximation for the sound absorption of these furnishings [18]. The sound absorption of an occupant is approximately equal to 0.55 m^2 (6.0 ft^2) for university students [32]. We should not include sound absorption of learning-space occupants again in any calculations because sound absorption of the occupants inside classrooms was already considered in the reverberation time values showed in Table 2 above.

7.9 Classroom Ambient Noise Levels

Classrooms are usually designed for speech communication. Unfortunately, classrooms are often not designed to optimally support this intended use. Ignoring acoustical design issues from the beginning will result in inaccurate speech communication. The cost of acoustical treatment needed after that will exceed the cost of doing the required optimal acoustical classroom design from the beginning. And of course, classrooms with poor acoustical designs will affect the learning process and the educational development.

Speech intelligibility in classrooms extensively degrades if there is an excessive ambient noise level and inadequate room acoustics design [24]. Usually listeners can guess correct spoken words in noisy environment without difficulty because of their quite remarkable abilities to guess. However, when the noise stops, a great difference in communication situation is noticed. Listeners with hearing impairment will face more difficulty in communicating in improbably acoustically designed places that have excessive noise. Table 3 shows the value of optimum reverberation times along with the maximum ambient noise levels to obtain optimum speech intelligibility conditions inside different classrooms. The louder the speech signals, compared to the level of interfering noise, i.e. high signal to noise ratio, the more intelligible the speech will be. Important sources of noise in classroom spaces are heating, ventilating and air conditioning (HVAC) equipment. In smart classrooms, projectors and computer devices are another source of noise. Students inside classroom themselves, also add unwanted noise that will interfere with spoken words and hence decreases the ability to understand speech.

8. Evaluating Measuring and Classroom Noise

Usually we measure Sound Pressure Level (SPL) for the octave band which gives us a noise spectrum. This noise spectrum is used to obtain Noise Criterion (NC), common used in U.S., or Noise Rating (NR), commonly used in Europe, by plotting the measured octave band levels on a set of NC or NR curves.

8.1 Noise Criterion – NC

NC is a single numerical index commonly used to define design goals for the maximum allowable background noise in a given space. It was established in United States to rate indoor noise, heating, ventilating and air conditioning noise. The NC criteria curves consist of a family of curves that define the maximum allowable octave-band sound pressure level corresponding to a chosen NC design goal [32]. As an example, a private office space would require a lower NC rating than a lobby area. NC curves extend from 63Hz to 8000 Hz to define the limits of octave band spectra, must not be exceeded to achieve the acceptance level of certain space, as shown in Figure 5. The measured noise spectrum is plotted to represent the measured NC curve, the lowest NC curve which is not exceeded by the measured NC, is the NC rating of this measure.

Table 3. Maximum ambient noise levels and optimum reverberation times (RT) for good speech intelligibility [18].

Example Situations	Maximum noise		RT
	dBA	NC	
Primary school classroom Boardroom for elderly adults	30	23	0.5
High school classroom General meeting room	35	28	0.7
Large lecture hall theatre	30	23	0.7

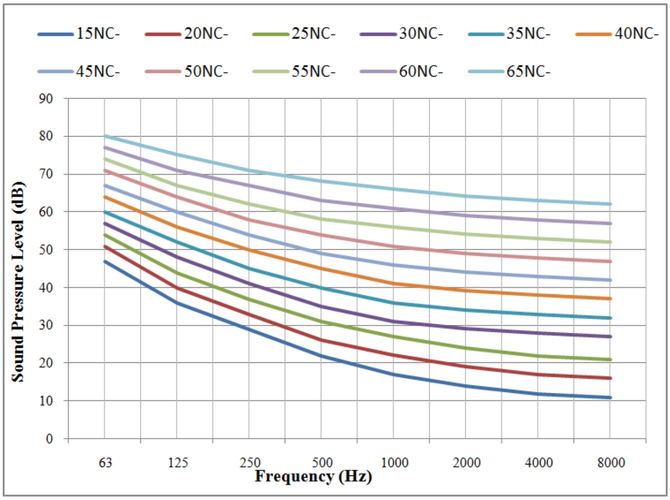


Fig. 5. Noise Criterion Curves [32].

8.2 Recommended NC For Classrooms

Recommended NC levels for different KAU room/space types should not exceed the NC limits listed in Table 4.

Table 4. Recommended NC levels for different KAU room/space types [32].

Type of Room - Space Type	Recommended NC Level NC Curve	Equivalent Sound Level (dBA)
Lecture and classrooms	**25-30**	**35-40**
Open-plan classrooms	**35-40**	**45-50**
Movie motion picture theaters	30-35	40-45
Libraries	35-40	40-50
Legitimate theaters	20-25	30-65
Private Residences	25-35	35-45
Restaurants	40-45	50-55
TV Broadcast studies	15-25	25-35
Recording Studios	15-20	25-30
Concert and recital halls	15-20	25-30
Sport Coliseums	45-55	55-65
Sound broadcasting	15-20	25-30

8.3 Noise Rating (NR)

Noise level can also be measured using Noise Rating (NR) set of curves developed by the International Organization for Standardization

(ISO) [27]. These curves are used to find the acceptable indoor environment for hearing preservation, speech communication and annoyance. In Europe it is common to use the NR. Unlike NC curves, different NR curves are obtained for each room type usage(s). Each curve has an NR number that represents the acceptable sound pressure level based on the room usage. Figure 6 shows NR curves for different sound pressure levels plotted over the octave band spectra [27].

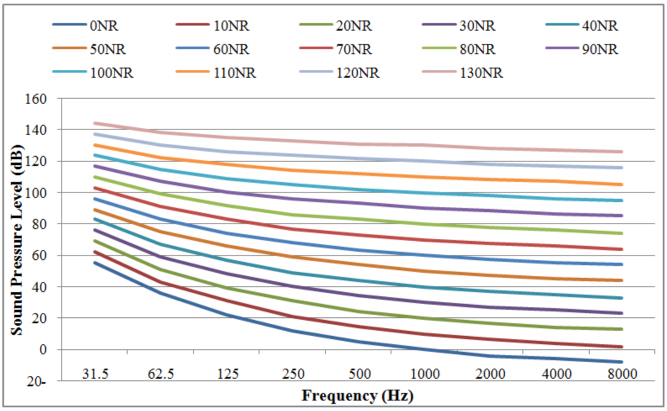


Fig. 6. Noise rating curves for different sound pressure levels plotted over the octave band spectra [27].

8.4 Recommended Noise Rating Levels

Noise rating levels for different room/space usages must not exceed the NR values defined in Table 5.

Table 5. Recommended noise rating levels for different room/space usages [35].

Noise rating curve	Application
NR 25	Concert halls, broadcasting and recording studios, churches
NR 30	Private dwellings, hospitals, theatres, cinemas, **conference rooms**
NR 35	**Libraries**, museums, court rooms, **schools**, hospitals operating theaters and wards, flats, hotels, executive offices
NR 40	Halls, corridors, cloakrooms, restaurants, night clubs, **offices**, shops
NR 45	Department stores, supermarkets, canteens, general offices
NR 50	Typing pools, offices with business machines
NR 60	Light engineering works
NR 70	Foundries, heavy engineering works

The noise rating graphs for different sound pressure levels are plotted at acceptable sound pressure levels at different frequencies [27]. Acceptable sound pressure level varies with the room and its use. Different curves are obtained for each type of use. Each curve is obtained by an NR number.

9. Personal Computer Noise Emissions

As identified by the Basic Criteria for Award of the Environmental Label (the Blue Angel) [34], the evaluation of noise emissions is based on the indication of the declared A-weighted sound power level in dB to the first decimal place. A-weighted sound power levels $L_{WA(1...3)}$ are determined and calculated on the basis of ISO/FDIS 7779:2010[35]. It must be made sure that in the case of different configurations of identically constructed units, the measurements are performed on the loudest individual components. Measurements shall be taken in the following operational modes:

1. The unit is in idle mode. The measurement of the $L_{WA(1)}$ shall be performed in accordance with ISO/FDIS 7779:2010[35] in the operating mode according to ECMA-74:2008 [30], Annex C.15.3.1. The measurement can be dropped if no fans are installed (*e.g*. CPU fans, power supply fans, computer system fans).
2. The hard-disk drive is enabled. The measurement of the $L_{WA(2)}$ shall be performed in accordance with ISO/FDIS 7779:2010 [35] in the operational mode according to ECMA-74:2008 [36], Annex C.9.3.2. The measurement can be dropped if no mechanical hard disk drive is installed.
3. An optical drive in a typical configuration is enabled. The measurement of the $L_{WA(3)}$ shall be performed in accordance with ISO/FDIS 7779:2010 in the operational mode according to ECMA-74:2008, Annex C.19.3.2. The measurement can be dropped if no optical drive is installed.

To make sure that the sound power level can be considered to declare as minimum of three devices need to be tested in each operational mode in accordance with ISO 9296:1988. The declared sound power levels $L_{WAd(1...3)}$ shall be determined on the basis of ISO 9296:1988 and given in dB to the first decimal place.

Alternatively, if the noise measurements can only be performed on one device the declared A-weighted sound power level L_{WAd} may be determined using the following formula following ISO 9296:1988:

$L_{WAd} = L_{WAE} + 3$ dB
(**L_{WAE}**= sound power level determined by means of a single measurement in dB Measurement conditions)
The values of the declared A-weighted sound power level **$L_{WAd(1...3)}$** recorded therein shall not exceed the following operational mode:

1. Hard disk drive enabled (HDD) $L_{WAd(1)}$ 42.0 dB
2. Idle mode $L_{WAd(2)}$ 38.0 dB
3. Optical drive enabled (ODD) $L_{WAd(3)}$ 50.0 dB

For measurement purposes, the Equipment Under Test (EUT) is placed on a test table according to DIN EN ISO 7779: 11-2002, arrangement with 9 measurement points as shown in Figure 7.
where:

I1, I2, I3 ≡EUT dimensions
d ≡ Measurement distance = 1.0 m
a = 0.5 I1 + d
b = 0.5 I2 +d
c = I3 + d
$S \equiv$ *Measuring surface* $= 4(ab + bc + ca)$

$$Surface\ sound\ value = 10\ log\frac{s}{s_0}$$

S ≡ Measuring surface
S0 ≡ Reference area = 1.0 m²

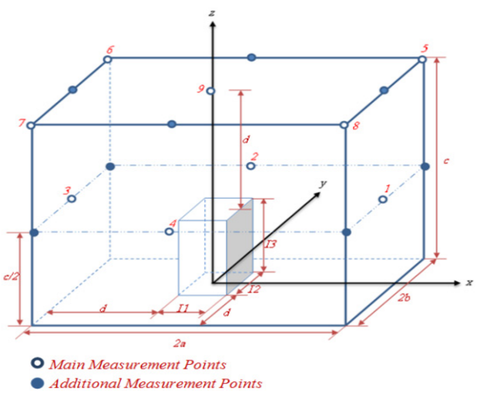


Fig. 7. Test arrangement for measuring noise emission from PC.

Measurements are done in anechoic chamber as shown in Figure 8, with the following measuring conditions:
Background noise level 14, 0 dB
Room correction K2 = 1, 33 dB
Measurement uncertainty 3, 0 dB
Absorption factor α = 0, 5
Length 10, 0 m
Width 10, 0 m
Height 5, 0 m
Reference area A = 200, 00 m²
Noise emission values according to ISO 9296: 1988,
ODD sound pressure level = 24.4 dB (A)
HDD sound pressure level = 22.3 dB (A)
Idle mode sound pressure level = 19.4 dB (A)
And according to RAL-UZ 78: 06-2006, "The Blue angel":
ODD Measured $\mathbf{L_{WAd}}$ = 39.9 dB (A)
HDD Measured $\mathbf{L_{WAd}}$ = 37.8 dB (A)
Idle mode Measured $\mathbf{L_{WAd}}$ = 34.9 dB (A)
$\mathbf{L_{WAd}} = \mathbf{L_{WAE}}$ + 3dB (Measurement uncertainty)

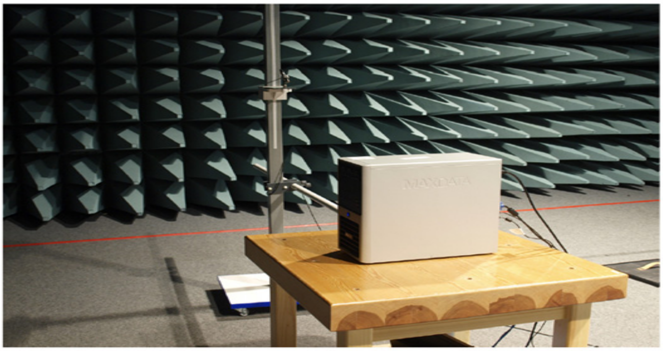


Fig. 8. Anechoic chamber for measuring noise emission from PC.

10. Conclusion

The work explained in this paper suggested a developed smart classroom acoustics design model (SCADM) that can be used by architects, acoustics engineers and designers in an early stage of classroom design in order to achieve the acoustical optimum conditions of KAU classrooms. The work tackled several important problems caused by the noise generated by students and instructional equipment in smart classroom that can seriously degrade communication and speech intelligibility and hence degrading learning quality. This type of interior noise source has been taken into consideration in the evaluation, to aid in the application of practical noise control measures. These measures may take the form of using neoprene chair-leg tips to minimize the sound of scuffling chairs and avoiding locating noisy projectors close to students. Excessive noise and reverberation are significant barriers to effective learning, particularly for students with special learning needs or hearing disabilities. A signal to noise ratio (SNR) of +15 dB is critical for an optimal learning environment. This can be addressed by increasing the sound quality inside the classroom by using sound field amplification but care must be taken here since this is not necessarily helpful if reverberation is excessive. Excessive noise in the classroom contributes to vocal strain in teachers and this can be managed if vocal and speech strategies for teachers are maintained. Classroom noise and acoustics need to be properly addressed during the design, building and retrofitting of the University classrooms. Increasing awareness of classroom acoustics and noise issues among architects, school officials and administrators is a must. New classroom must be located away from exterior noise sources, and design new classroom with the appropriate room size, ceiling height, and sound insulation material. Also, appropriate use of absorptive and reflective materials is important to keep reverberation times within optimal rang for classrooms. By achieving the quality and efficiency of the smart classroom and educational environment, we can reach excellent speech intelligibility, and hence increasing students learning outcomes.

11. Acknowledgment

Authors would like to thank King Abdulaziz University in Jeddah, Saudi Arabia for offering the necessary facilities for accomplishing this research.

References

[1] **Smith, B. N.,** "Noise assessment", Acoustic Associate, pp: 1-20, February, (2014).

[2] **Jensen, Jesper Taal, Cees H**., "Speech Intelligibility Prediction Based on Mutual Information", *IEEE/ACM transactions on audio, speech, and language processing*, **22**, (2), February, (2014).

[3] **Hygge, S.,** "Classroom noise and its effect on learning", *11th International Congress on Noise as a Public Health Problem (ICBEN),* Nara, JAPAN, June 4th, (2014).

[4] **Curiale, L., *et al*.,** "Classroom Acoustics Research", *http://www.vaniercollege.qc.ca/ student-research-centre/files/ 2014/05/classroom-acoustics-2013-2014.pdf.*

[5] **Kristiansen, J., *et al.,*** "A study of classroom acoustics and school teachers' noise exposure, voice load and speaking time during teaching, and the effects on vocal and mental fatigue development", *International Archives of Occupational and Environmental Health*, November, **87**, (Issue 8), pp: 851-860, Jan, (2014).

[6] **Sodsri, C.,** "Effects of classroom reverberation and listeners' locations to speech intelligibility", *Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 9th International Conference*, 16-18 May, (2012).

[7] **Wang, J., Yan, N., Wang, D.** and **Peng, J.,** "Comparison of objective acoustical indices in different classroom environments in primary school", *The 21st International Congress on Sound and Vibration*, Beijing/China, 13-17 July, (2014).

[8] **Klatte, M., Hellbrück, J., Seidel, J.** and **Leistner, P.,** "Effects of Classroom Acoustics on Performance and Well-Being in Elementary School Children: A Field Study", *Environment and Behavior*, **42**, 659, (2010).

[9] **Dockrell, J.E.** and **Shield, B.M.,** "Acoustical barriers in classrooms: the impact of noise on performance in the classroom", *British Educational Research Journal*, **32**, (3), pp: 509-525, (2006).

[10] **Wall, K., Dockrell, J.** and **Peacey, N.,** "The importance of the built environment for learning- a research evidence overview", *Proceedings of 39th International Congress on Noise Control Engineering INTER-NOISE* Lisbon, June, 13-16, p. 453, (2010).

[11] **Iannace, G., Ciaburro, G.** and **Maffei, L.,** "Effects of shared noise control activities in two primary schools", *Proceedings of 39th International Congress on Noise Control Engineering INTER-NOISE Lisbon*, June, 13-16, p.524, (2010).

[12] **Wróblewska, D.** and **Leo, K.,** "Influence of Acoustical Adaptation on Classroom's Acoustical Environment", *Acta Physical Polonica* A (2012).

[13] **Julie, D.** and **Bridget, S.,** "The impact of sound field systems on learning and attention in elementary school classrooms", *Journal of Speech, Language, and Hearing Research*, ISSN 1092-4388 (print); 1558-9102 (electronic), (2012).

[14] **Rand, I.,** "Achieving acoustical standards in the classroom, *Study of HVAC systems and classroom acoustics*", Trane, (2011).

[15] **Henrique, P., Zannin, T., Zwirtes, D. P. Z.** and **Passero, C. R. M.,** "Assessment of Acoustic Quality in Classrooms Based on Measurements, Perception and Noise Control", *Noise Control, Reduction and Cancellation Solutions in Engineering Chapter 10*, Mar (2012).

[16] **Yao, Z.** and **Peng, Z.** "The Research of Acoustic Indicators Defects and Solutions of the College Classrooms, take Shenyang Jianzhu University as an example", *Journal of Shenyang Jianzhu University (Social Science)*, (2010).

[17] **Zannin, P.H.T.** and **Zwirtes, D.P.Z.,** "Evaluation of the acoustic performance of classrooms in public schools", *Applied Acoustics*, **70**, (Issue 4), pp: 626-635, April (2009).

[18] **American National Standard,** ANSI S12.60: *Acoustical performance criteria, design requirements, and guidelines for schools*, Melville, (2002).

[19] **Thomas, J.,** "Smart E-classrooms, Traditional Classrooms and Critical Thinking", *World Conference on E-Learningin Corporate, Government, Healthcare, and Higher Education*, AACE, Vol. **2002**, (Issue 1), pp: 2288-2291, (2002).

[20] **Acoustical Society of America,** "Classroom Acoustics: A Resource for Creating Learning Environments with Desirable Listening Conditions", Available at *http://www.nonoise.org/quietnet/qc/booklet.pdf*; accessed August (2014).

[21] **Tanner, C. K.,** "Minimum Classroom Size and Number of Students Per Classroom", *The University of Georgia School Design and Planning Laboratory*, April, (2000).

[22] **Tanner, C. K.,** "Explaining Relationships Among Student Outcomes and the School's Physical Environment", *Journal of Advanced Academy*, JAA, **19**, (3): 444–471, Spring (2008).

[23] **Lecture theatre design** - Bath University - Part II, available at: *http://machacoustics.blogspot.com/2010/11/lecture-theatre-design-bath-university.html*, accessed October (2014).

[24] **Noaman, A. Y., Ragab, A. H. M.** and **Madbouly, A. I.,** ”CAAM: A Novel Classroom Acoustics Assessment Model for Enhancing Learning Quality (Case Study: KAU)”, *IDEE* (2013).

[25] **Calculations, S.,** Adding of equal loud incoherent sound sources available at:*http:// www. sengpielaudio.com/calculator-leveladding.htm*, accessed October (2014).

[26] **Squared, M.C., System Design Group,** Inc., "So why does reverberation affect speech intelligibility*?". http://www.mcsquared.com/y-reverb.htm*, Retrieved (2013).

[27] **Sabine, W. C.,** “Reverberation,” *in Collected papers on acoustics.* Harvard university press, (1922).

[28] **Bradley, J.S.,** “Acoustical Design of Rooms for Speech ", Construction Technology Update No. 51, published by *Institute for Research in Construction*, ISSN 1206-1220,March (2002).

[29] **Bradley, J. S., Sato, H.** and **M. Picard,**"On the importance of early reflections for speech in rooms". *The Journal of the Acoustical Society of America*, **113** (6): 3233-3244, (2003).

[30] **Acoustical Society of America**, "Classroom Acoustics: A Resource for Creating Learning Environments with Desirable Listening Conditions", Available at *http://www.nonoise.org/quietnet/qc/booklet.pdf; accessed August* (2012).

[31] **How to Solve a Flutter Echo Problem**, *http://www.acousticalsolutions.com/how-to-solve-a-flutter-echo-problem*, accesses october (2014).

[32] **Gary W. Siebein, Martin A. Gold ,Glenn W. Siebein, and Michael G. Ermann**," Ten Ways to Provide a High-Quality Acoustical Environment in Schools", *Language, speech, and hearing services in schools* • Vol. 31 pp 376–384 , October 2000.

[34] **Seddeq, H. S.,** " Factors Influencing Acoustic Performance of Sound Absorptive Materials", *Australian Journal of Basic and Applied Sciences*, **3** (4): 4610-4617, (2009).

[35] **ISO/FDIS 7779**:2010 Acoustics — “Measurement of airborne noise emitted by information technology and Telecommunications” , *https://shop.austrian- standards.at/ Preview.action*, (2010).

[36] **CRI Technical Services**, Carpet and High Performance Schools, January 2003. *http://www.canadiancarpet. org/carpet_in_schools/pdf/* ECMA-74:2008,“Measurement of Airborne Noise Emitted by Information Technology and Telecommunications Equipment”, *www.ecma-international.org/publications/files/ECMA-ST/ECMA-74.pdf*, (2003).

نموذج مطور لتصميم الصوتيات داخل الفصول الدراسية الذكية لتحسين البيئة التعليمية

عبدالحميد محمد رجب، وأمين يوسف نعمان، وأيمن إسماعيل مدبولي

كلية الحاسبات وتقنية المعلومات، جامعة الملك عبدالعزيز، جدة، المملكة العربية السعودية

aragab@kau.edu.sa

المستخلص. تعتمد جودة وكفاءة عمليات التعليم والتدريس في الجامعات الأكاديمية على العديد من العوامل المهمة. إذ يعتبر جودة الصوت داخل بيئة التعلم في الفصول الدراسية ومحيطها، هي واحدة من أهم هذه العوامل التي تؤثر على جودة العملية التعليمية. وقد ركزت العديد من الدراسات والأبحاث حديثًا، على موضوع الراحة الصوتية في الفصول الدراسية الجامعية، ومدى تأثيرها على العملية التعليمية.

ندرس في هذا البحث خصائص تصميم الصوتيات داخل الفصول الدراسية الذكية، واللازمة لتحقيق شروط الصوتيات ذات الجودة العالية. وتشمل هذه الخصائص على القضايا المتعلقة بالتصميم داخل وخارج مصادر الضوضاء، والتصميم الأمثل للصدى الزمني، واختيار مواد عزل الصوت ومواد المعالجة الصوتية، والتصميم لوضوح الكلام، والاستماع داخل الفصول الدراسية بجامعة الملك عبدالعزيز.

حيث يقدم البحث اقتراح نموذج مطور لتصميم الصوتيات للفصول الدراسية الذكية، التي يمكن استخدامها من قبل المهندسين المعماريين، ومهندسين الصوتيات والمصممين في مرحلة مبكرة من تصميم الفصول الدراسية، من أجل تحقيق شروط الصوتيات داخل الفصول الدراسية الجامعية، بجامعة الملك عبدالعزيز. وذلك بهدف رفع مستوى جودة وكفاءة البيئة التعليمية الجامعية، للوصول إلى بيئة تعليمية ممتازة، تؤدى إلى تحسين مستوى مخرجات التعليم لدى الطلاب.

مجلة جامعة الملك عبدالعزيز: علوم الحاسبات وتقنية المعلومات، م٢، ١٢٨ صفحة (٢٠١٣م/١٤٣٥هـ)
ردمد ٦٣٣٦-١٦٥٨
رقم الإيداع ١٤٣٤/١٠٢٨

مجلة
جامعة الملك عبد العزيز
علوم الحاسبات وتقنية المعلومات

المجلد ٢
٢٠١٣م
١٤٣٥هـ

مركز النشر العلمي
جامعة الملك عبد العزيز
ص.ب. ٨٠٢٠٠- جدة ٢١٥٨٩
http://spc.kau.edu.sa

■ سعر النسخة ■

- داخل المملكة ١٠ ريالات سعودية
- خارج المملكة ١٠ دولارات أمريكية

■ البيع والاشتراك ■

مركز النشر العلمي – جامعة الملك عبدالعزيز
ص.ب. ٨٠٢٠٠- جدة ٢١٥٨٩- المملكة العربية السعودية

■ التبادل ■

عمادة شئون المكتبات – جامعة الملك عبدالعزيز
ص.ب. ٨٠٢١٣- جدة ٢١٥٨٩- المملكة العربية السعودية

مطابع جامعة الملك عبدالعزيز

المحتويات

﴿ القسم الإنجليزي ﴾

صفحة

بسم الله الرحمن الرحيم