An Efficient Key Management Scheme for Secure Wireless Multicast

Yan Sun, Wade Trappe, and K. J. Ray Liu
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742

***Abstract*—In the future, many multicast services will take place in the wireless domain. However, before these services can successfully be deployed, security infrastructures must be developed that manage the keys needed to provide access control to content. In this paper, we present a method for designing multicast key management trees that are suitable for mobile wireless environments. By matching the key management tree to the cellular network topology, the total communication burden is reduced by 33%-45% compared to using the traditional key management trees that are independent of the topology.**

I. Introduction

The advancements in wireless technologies promise to free users from the confines of static communication networks. Users will be able to work, shop, and be entertained from anywhere at anytime. There has also been significant progress in both the technology underlying multicast networking as well as the deployment of applications utilizing multicast technologies. Already there are services using multicast which stream stock quotes, and provide video and audio on demand. It is reasonable to forecast that consumers will desire to have a similar suite of applications running on their portable devices, especially as technologies such as 3G are successfully installed.

These applications will require mechanisms to provide access control to multicast content. Access control is typically provided through encryption, which requires the maintenance and distribution of keying information. A popular class of multicast key management schemes are those that employ a tree hierarchy for maintenance of keying material[1][2][3]. These schemes focus entirely on the problem of dynamic membership, where users join or leave the multicast service. Tree-based schemes tend to have desirable usage of computation, communication, and storage resources for the user and the group controller. They do not, however, consider issues related to the delivery of rekeying messages, and do not consider the underlying network topology.

In this paper, we propose a method for designing the multicast key management tree for a group of mobile users in a cellular network. By matching the key management tree to the network topology, we reduce the communication burden associated with rekeying. In Section II, we introduce the concept of matching the key tree to the network topology and motivate the reduction of the communication burden of the rekeying messages. In mobile environments, the user will subscribe to a multicast service under an initial host agent, and through the course of his service move to different cells and undergo *handoff* to different base stations. Although the user has moved, he maintains his subscription to the multicast group. Therefore, it is important to address issues arising from user relocation for the topology matching key management tree. In Section III, we present a handoff scheme that is suitable for topology matching key management trees. We describe, in Section IV, a tree structure that can easily adapt to changes in the number of users, and can be used to build a key tree that matches the network topology. We then describe how to choose the parameters that optimize the tree. Finally, simulation results are presented in Section VI and conclusions are described in Section VII.

II. Topology-Matching Key Management Tree

In this section, the basic ideas of the Topology Matching Key Management scheme are introduced. Access control for multicast application typically employs a tree of encryption keys that are used to update and maintain a key, known as the *session key*, that is shared by all group members[1][2][3]. In such schemes, when a user leaves the service, it is necessary to change keys associated with the departing member in order to prevent him from accessing future group data. Similarly, when a user joins, it is necessary to update keys in order to prevent a joining user from accessing past content. In tree-based multicast key management schemes, most rekeying messages are only useful to a subset of users, who are always neighbors on the key management tree. For example, Figure 1 shows a key management tree with 16 users. Assume user 1 is leaving, then 5 messages need to be sent to update K_{111}, K_{11}, K_1, K_ε and K_s respectively. The 1st message, used to update K_{111}, is only useful to user 2. The 2nd message is only useful to user 2,3, and 4. Similarly, the 3rd message is only useful to user 2,3,4,$\cdots$,7. The 4th and the 5th messages are useful to all users [3]. Therefore, rekeying messages do not have to be sent to every user.

We design a key management tree that matches the network topology in such a way that the neighbors on the key tree are also physical neighbors on the network. Additionally, by delivering the rekeying messages only to the users who need them, we may take advantage of the fact that the key tree matches the network topology, and localize the delivery of rekeying messages to small regions of the network. This lessens the amount of traffic crossing portions of the network that do not have users who need to be rekeyed. In order to accomplish this, it is necessary to have the assistance of entities that control the rekeying message transmission, such as the BS's in cellular wireless network.

A cellular network model, proposed in [4], consists of mobile users, base stations (BS) and supervisor hosts (SH) (Figure 2).

0-7803-7400-2/02/$17.00 © 2002 IEEE

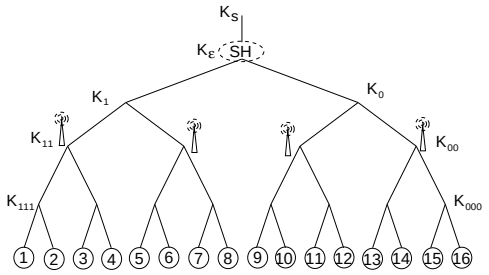


Fig. 1. A typical Key Management Tree

SH handle most of the routing and protocol details for mobile users, and is part of the wired network. In this work, we assume that there is only one SH responsible for administering the BS's, and for managing the keys necessary to protect the multicast communications. It is also assumed that users are uniformly distributed under the BS's, and the number of users is large. The users under one BS can be looked at as a subgroup, and we assume that a BS knows whether a rekeying message is needed by its subgroup. Therefore, a rekeying message is first multicast to all BS's by the SH through wired connections, then the BS's broadcast this message if it is useful to their subgroups. By doing this, we only send rekeying messages to the subgroup(s) needing the message, instead of to all users.

Based on the discussion above, we design a key management tree that matches that network topology in two steps:

1) Design a subtree for the users under each BS. Those subtrees are called *user-subtrees*.
2) Design a subtree which governs the key hierarchy between the BS's and the SH and shall be called *BS-subtree*.

Since the combined key management tree depends on the network structure, we call it a Topology-Matching Key Management (TMKM) tree. For example, the tree shown in Figure 1 is a TMKM tree for the network topology shown in Figure 2.

Traditional key management trees[1][2][3] are independent of the network structure, and we call them Topology Independent Key Management(TIKM) trees. When using a TIKM tree, rekeying messages are sent to every user, i.e. broadcast by all BS's. When using a TMKM tree, rekeying messages are broadcast by only a subset of BS's. Let S_1 denote the number of the messages multicast to the BS's, and S_2 denote the number of the messages broadcast by the BS's. For example, if one message is multicast to all BS's and then broadcast by 2 BS's, then $S_1 = 1$ and $S_2 = 2$. The measurement of communication burden, wire-line cost C_{wire}, wireless cost $C_{wireless}$, and total cost C_T, are defined as:

$$\begin{aligned} C_{wire} &= E[S_1]\,; \quad C_{wireless} = E[S_2] \\ C_T &= \gamma \cdot C_{wireless} + (1-\gamma) \cdot C_{wire} \end{aligned}$$

where γ ($0 \le \gamma \le 1$) is the wireless weight, which indicates the importance of considering the wireless cost. Given the wireless weight, both the TMKM and TIKM trees should be designed to minimize the total communication cost, C_T.

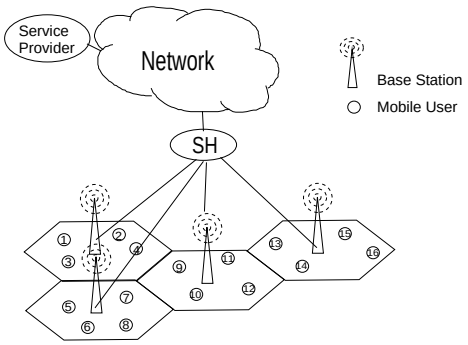


Fig. 2. A Cellular Network Model

III. Hand-off Schemes for TMKM trees

In the wireless scenario, the multicast hand-off problem usually refers to issues of data flow, in which packets are lost or duplicated as users move from cell to cell. When using the TMKM tree, the hand-off problem involves not only data flow, but also key management. Since the TMKM tree depends on the network structure, when a user moves from one cell to another cell, the user needs to be moved from one branch to another branch of the TMKM tree. Moving users on the tree causes extra communication, which is the major drawback of the TMKM tree. In the sequel, the expression *hand-off scheme* only refers to the process of moving a user on the TMKM tree. In this section, we analyze the hand-off problem and then propose an efficient hand-off scheme for the TMKM tree.

In most cases, mobile devices with one transceiver can only establish communication to one BS. In this work, we do not assume that a user only gets messages from one BS at a time, but that users may exchange information amongst themselves (perhaps, to collude in hopes of compromising other users' keys). The proposed hand-off scheme guarantees security even if users can listen to the communication of any cell at any time.

First of all, let's explain a simple solution. When a user α moves from cell i to cell j, we need to:

1) Update the set of keys user α previously had using a regular member leave procedure, e.g.[3]. This operation is later referred to as "remove user α from cell i".
2) Choose a branch of the subtree under BS j, where the users will be placed, and perform a member join operation, e.g.[3], to update the necessary keys. This operation is later referred to as "add user α to cell j ".

This scheme is not practical for mobile network with frequent hand-offs because the extra communication cost is too high. Instead, by allowing a user to have more than one set of valid keys when he stays in the service and update all of his keys when he leaves the multicast service, an efficient hand-off scheme is designed as:

(A) When user α moves from cell i to cell j:

1) α is put on the WTBR (wait to be removed) list of cell i. Each BS has a WTBR list stored at the SH.
2) Choose a branch of the subtree under cell j, which was most recently updated at time T_0 due to other users' leav-

ing. Let T_1 denote the time when α joined the multicast service. If T_1 is earlier than T_0, a set of keys associated with this branch needs to be updated using the general user join procedure. If T_1 is later than T_0, no update is needed.

3) The set of keys for that branch is sent to α through a unicast channel.

(B) When user α leave the multicast service from cell j:

1) Cell j, and all the cells whose WTBR lists contain α need to update the corresponding keys. Updating those keys together is more efficient than individually.
2) Check the WTBR list of cell j and find all the users who previously belonged to the same branch as α does. Those users are removed from the WTBR list without any extra cost.
3) Check other WTBR lists containing α and find all users previously belonging to the same branch as α previously did. Those users are removed from the WTBR lists without extra cost.
4) Remove α from all WTBR lists.

Since more than one set of keys may be updated when a user leaves, handoff results in extra communication cost, which will be calculated in Section V.

IV. Key Management Subtree Design

In this section, we discuss the design of the *user-subtrees* for the users under one BS. Many key management trees have been proposed that attempt to minimize the rekeying message size when a user joins/leaves the multicast service. However, only a few papers [5] have addressed the problem of maintaining the desired properties of the tree, such as small rekeying message size, after users join/leave. Based on the user join/leave procedure described in [3], we designed an $(a, L, \mathbf{x})$-logic tree (Figure 3), which maintains the tree structure all the time and can be optimized based on the statistics of the number of users.

The $(a, L, \mathbf{x})$-logic tree has $L+1$ levels. The upper L levels, which is a symmetric subtree with degree a, are fixed during the multicast service. The $(L+1)^{th}$ level changes when users join/leave. Users are attached to the upper nodes randomly. We use a vector $\mathbf{x}$ to describe this level, where x_i is the number of users attached to the ith node, $i = 1, 2, \cdots, a^L$. For example, in Figure 3, $\mathbf{x} = [4, 2, 3, 3, 2, 4, 3, 3, 3]$. This $(a, L, \mathbf{x})$-logic tree, which we call the ALX tree, maintains the tree structure when users join/leave.

Next, we will analyze the performance of the ALX tree by comparing it with fixed degree trees. The performance criteria is the expected value of the rekeying message size. To simplify the analysis, the following assumptions are made:

1) The user's arrival process is Poisson with rate λ.
2) The period of time a user stays in the multicast service, referred to as the service time, is an exponential random variable with mean $1/\mu$.

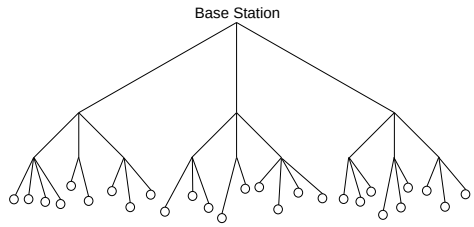


Fig. 3. ALX Tree

3) $\{x_i\},\ i = 1, 2, \cdots, a^L$ are i.i.d. Users' join/leave behavior are independent of each other.

Let k denote the number of users in the multicast service. Based on the first two assumptions, k is governed by a Poisson random variable K with rate θ i.e. $p(k) = \frac{\theta^k}{k!}e^{-\theta},$ where $\theta = \lambda/\mu$.

We can show that the communication cost of the ALX tree, C_{alx}, is:

$$\begin{aligned} C_{alx} &= E\left[\begin{array}{c}\text{rekeying message size sent for a}\\ \text{ALX tree with parameters } a, L\end{array}\right] \\ &= \left(\sum_{k=1}^{\infty} p(k)\cdot k\cdot\left(\frac{k}{a^L}-1+aL\right)\right)\cdot\mu \qquad (1)\end{aligned}$$

Then, the optimization problem can be defined as:

$$\tilde{C}_{alx} = \min_{a,L} C_{alx}\ , \quad \text{where } a, L \text{ are positive integers.}$$

We can also derive the performance lower bound for the key management trees with fixed degree n as:

$$\begin{aligned} C_{fix} &= E\left[\begin{array}{c}\text{rekeying message size sent}\\ \text{for a tree with degree n}\end{array}\right] \\ &> \sum_{k=1}^{\infty} p(k)\cdot k\cdot\mu\cdot(n\log_n(k)-1)\end{aligned}$$

This bound cannot be achieved by a fixed degree tree, and is used as a reference to evaluate the ALX tree. Let $\tilde{C}_{fix} = \min_n C_{fix}$. In Figure 4, $\tilde{C}_{fix}$ and $\tilde{C}_{alx}$ are compared for different user join rates, λ. We can see that the optimized communication cost $\tilde{C}_{alx}$ for the ALX tree is very close to the lower bound for the communication cost of a fixed degree tree. In this paper, ALX tree structure is also used to design the *BS-subtrees*, and TIKM trees.

V. TMKM tree Design and Optimization

In this section, we derive the communication cost for the TMKM tree, and describe the procedure for designing the TMKM tree. We first define several variables. The random variable K denotes the number of users in the multicast service, and $p(k)$ denotes the pmf of K. The random variable I denotes the number of WTBR lists that contain a user when he leaves, and the pmf is $p_h(i), i = 1, 2, ..., n_{bs}$ and n_{bs} is the number of base stations. a and L are the degree and the level

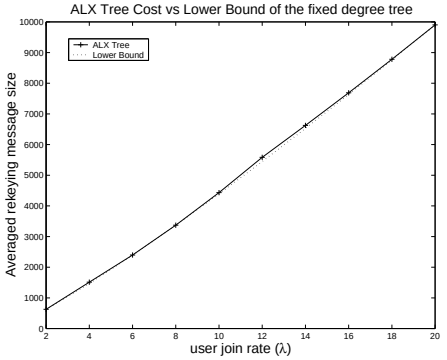


Fig. 4. ALX tree and lower bound

of the user-subtrees. a_{bs}, L_{bs} is the degree and the level of the BS-subtree. γ denotes the wireless weight.

Besides the three assumptions made in Section IV, we also assume:

1) The number of BS's is fixed, i.e. n_{bs} is a constant.
2) Users are uniformly distributed under BS's. Thus a and L are same for every subgroup.
3) I is independent of K.
4) Compared to the cost of user-leave, the cost of user-join can be neglected.

Then, the wireless cost, defined in Section II, is computed as:

$$C^{mkm}_{wireless} = \sum_{k=1}^{\infty} p(k) \cdot k \cdot \mu \cdot \left(\sum_{i=1}^{n_{bs}} p_h(i) \cdot T_{wireless} \right),$$

where $T_{wireless}$ is the expected value of S_2 given k users in the multicast service, 1 user wants to leave and he is on i WTBR lists. We can show that:

$$\begin{aligned} T_{wireless} \approx\ & (\frac{k/n_{bs}}{a^L} - 1 + aL) \cdot i + s^2 \cdot B(a_{bs}^{L_{bs}}, i) \\ & + \sum_{m=1}^{L_{bs}} a_{bs} \cdot a_{bs}^m \cdot s \cdot B(a_{bs}^{L_{bs}-m}, i) \quad , \end{aligned}$$

where $s = n_{bs}/a_{bs}^{L_{bs}}$, and $B(x, y)$ is the expected number of non-empty boxes when putting y items randomly into x boxes with repetition. Similarly, the wire-line cost is:

$$C^{mkm}_{wire} = \sum_{k=1}^{\infty} p(k) \cdot k \cdot \mu \cdot \left(\sum_{i=1}^{n_{bs}} p_h(i) \cdot T_{wire} \right),$$

where

$$\begin{aligned} T_{wire} =\ & (\frac{k/n_{bs}}{a^L} - 1 + aL) \cdot i + s \cdot B(a_{bs}^{L_{bs}}, i) \\ & + \sum_{m=1}^{L_{bs}} a_{bs} \cdot B(a_{bs}^{L_{bs}-m}, i) \quad . \end{aligned}$$

Then, the total cost is:

$$\begin{aligned} C_T^{mkm} &= (1-\gamma) \cdot C^{mkm}_{wire} + \gamma \cdot C^{mkm}_{wireless} \\ &= \left(\sum_{i=1}^{n_{bs}} p_h(i) \cdot i \right) \mu \cdot T_1 + \left(\sum_{k=1}^{\infty} p(k) \cdot k \right) \mu \cdot T_2 \ , \end{aligned}$$

where,

$$\begin{aligned} T_1 =\ & \sum_{k=1}^{\infty} p(k) \cdot k \cdot \left[\frac{k/n_{bs}}{a^L} - 1 + aL \right] \\ T_2 \approx\ & \sum_{i=1}^{n_{bs}} p_h(i) \cdot \Bigg(B(a_{bs}^{L_{bs}}, i) \left(s^2\gamma + s(1-\gamma) \right) \\ & + \sum_{m=1}^{L_{bs}} B(a_{bs}^{L_{bs}-m}, i) \cdot a_{bs} \left(a_{bs}^m s\gamma + 1 - \gamma \right) \Bigg) \quad . \end{aligned}$$

The TMKM tree needs to be optimized by choosing the parameters a, L, a_{bs} and L_{bs}, such that C_T^{mkm} is minimized. Because T_1 is a function of a and L, and T_2 is a function of a_{bs} and L_{bs}, the optimization problem can be decomposed into two subproblems:

$$\min_{a,L,a_{bs},L_{bs}} C_T^{mkm} \quad \Leftrightarrow \quad \min_{a,L} T_1 \text{ and } \min_{a_{bs},L_{bs}} T_2 \ .$$

The separability of the optimization problem allows for the TMKM tree to be designed in two steps:

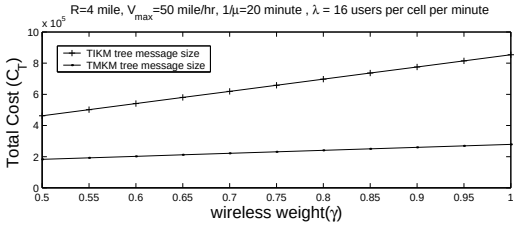
1) Construct the user-subtrees as a ALX tree using the parameters a and L which minimize T_1.
2) Construct the BS-subtree as another ALX tree with the parameters a_{bs} and L_{bs} that minimize T_2.

The optimal parameters can be found by performing a search on possible a, L, a_{bs}, L_{bs} values, which is feasible since the search space is small.

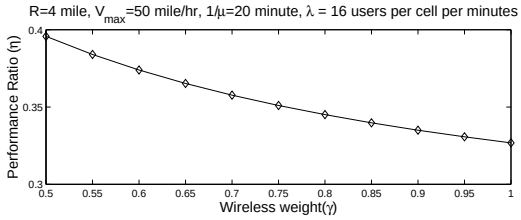
VI. Simulation Results

In this section, we compare the performance of the TMKM tree and the TIKM tree by both analysis and simulations. The system is described as follows:

- Similar to [6], we employ a homogeneous cellular network that consists of 12 concatenated cells. A ring network is used to avoid edge effects. We use the mobility model proposed in [7], where R is the radius of the cells, and V_{max} is the maximum speed of the mobile users.
- As discussed in Section IV, the user's arrival process is assumed to be Poisson with rate λ, and the service time is assumed to be an exponential random variable with mean $1/\mu$. These assumptions, which have long been used for cellular mobile telephone service [6][8], may not be accurate for multicast services. The user-join and service time model of the multicast, referred to as the service model, should depend on the type of multicast services. For example, the service model for movie multicast is different from that for periodic news multicast. We are not aware of any such models for multicast service, and for simplicity, the Poisson model was chosen in our analysis. The advantages of the TMKM tree over the TIKM tree do not seem to be sensitive to the choice of the service model.
- The wireless part should be assigned a larger weight than the wire-line part, i.e. $\gamma > 0.5$. Since the packet loss

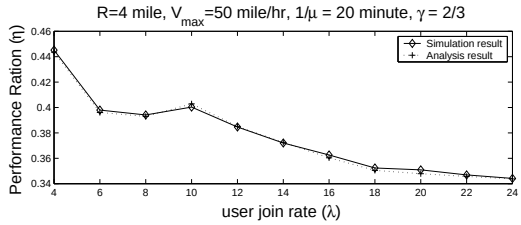


(a)

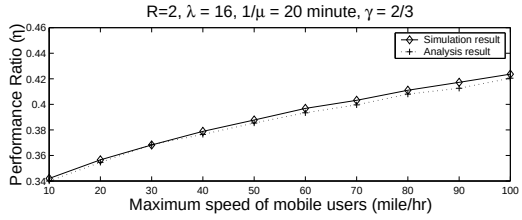


(b)

Fig. 5. (a) The total message size as the function of the wireless weight. (b) Performance Ratio as a function of the wireless weight.



(a)



(b)

Fig. 6. (a) Performance Ratio as a function of user join rate, (b) Performance Ratio as a function of users' maximum speed

rate of wireless transmission is typically higher than wired transmission, and the number of users under one BS is greater than the number of BS, it is important to place emphasis on the wireless component of the optimization.

- The TIKM tree is designed as an ALX tree. The wire-line cost, C_{wire}^{ikm}, can be computed using (1). Then, the wireless cost is computed as: $C_{wireless}^{ikm} = C_{wire}^{ikm} \times n_{bs}$.
- We define the *performance ratio* $\eta = C_T^{mkm}/C_T^{ikm}$. η is less than 1 and represents how much better the TMKM tree is compared to the TIKM tree. The smaller values of η correspond to TMKM trees with more advantages over TIKM trees.

In Figure 5, the TMKM trees are compared with the TIKM trees for different wireless weights, γ. Figure 5(a) shows the communication cost of the TMKM tree and the TIKM tree for a given set of parameters for the users' mobility and service models. The performance ratio is shown in Figure 5(b). Three observations are made. First, the TMKM tree cost is always less than 40% of the TIKM tree cost. Second, η is smaller for the larger wireless weight. This is because the TMKM tree tends to have larger wire-line cost and smaller wireless cost than the TIKM tree. This property can be shown by studying the cost functions derived in Section IV and V. Third, when $\gamma = 1$, the TMKM tree cost is as small as 33% of the TIKM tree cost. $\gamma = 1$ represents the cases when the wireless transmission is the bottle neck.

Figure 6(a) shows both the analysis and the simulation results of the performance ratio η for different user join rates, λ, with $\gamma = 2/3$. The advantage of the TMKM tree is larger, when the system contains more users. This property also can be verified by studying the cost functions derived in previous sections. Figure 6(b) shows the results for the different maximum speed of mobile users. As expected, the TMKM tree works better when the users move slower, which is the situation when less hand-offs occur.

VII. Conclusion

In this paper, we presented a method for designing the multicast key management tree for the mobile wireless environment. By matching the key management tree to the cellular network topology, a reduction in communication burden of the rekeying messages was observed compared to trees that are independent of the topology. It was shown that the problem of optimizing the communication cost for the TMKM tree is separable and can be solved by considering the wireless and the wire-line contributions separately. Simulations were performed for different user-join rates and mobile user speeds, and indicated that the cost of the TMKM tree was approximately 33-45% of the cost of the TIKM tree.

References

[1] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. on Networking*, vol. 8, pp. 16–30, Feb. 2000.

[2] R. Canetti, J. Garay, G. Itkis, D. Miccianancio, M., and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *INFOCOM'99*, 1999.

[3] W. Trappe, Jie Song, R. Poovendran, and K.J.R. Liu, "Key distribution for secure multimedia multicasts via data embedding," in *ICASSP'01*, May 2001.

[4] K. Brown and S. Singh, "Relm: Reliable multicast for mobile networks," *Computer Communication*, vol. 2.1, no. 16, pp. 1379–1400, June 1996.

[5] D. Balenson, D. McGrew, and A. Sherman, "Key management for large dynamic groups: one-way function trees and amortized initialization," Internet Draft Report.

[6] M. Rajaratnam and F. Takawira, "Nonclassical traffic modeling and performance analysis of cellular mobile networks with and without channel reservation," *IEEE Trans. on Vehicular Technology*, vol. 49, no. 3, pp. 817–834, 2000.

[7] M. M. Zonoozi and P. Dassanayake, "User mobility modeling and characterization of mobility patterns," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1239–1252, 1997.

[8] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. on Vehicular Technology*, vol. VT-35, no. 3, pp. 77–92, 1986.