Integrating Multicasting and Hash Algorithm to Support Host Mobility

Sahar Al-Talib, Borhanuddin Mohd Ali, and Sabira Khatun
Department of Communication and Networking System, Universiti Putra Malaysia, Malaysia

Abstract: *This paper explains a framework that has been proposed to construct a dynamic delivery tree for the Mobile Node (MN) movement in a mobile IPv6 network. The branches of the tree constitute the shortest paths from the packet source to each of the visited locations. The tree is dynamic so that the branches grow and shrink to reach the MN when necessary. This architecture is multicast-based, in which a mobile node is assigned a multicast address and the correspondent nodes send packets to that multicast group. As the mobile node moves to a new location, it joins the multicast group through the new location and prunes through the old location. Hash Algorithm has been implemented as a mechanism for the MN to join and leave a multicast group. Dynamics of the multicast tree provide for smooth handoff, efficient routing and conservation of network bandwidth. To allow a smooth handoff, the MN should not prune the old location until it starts receiving packets from the new location. The performance of the proposed mechanism was evaluated through a simulation model built for this purpose. The simulation results showed that the dynamics of joining and leaving the group directly affect handoff latency and smoothness, as a result it conserved Radio Frequency (RF) bandwidth.*

Key words: *Mobile IPv6, multicast group, hash algorithm, handover.*

Received July 27, 2004; accepted December 28, 2004

1. Introduction

Multicast is a mechanism for efficient one-to-many communication in which the source transmits a single packet and the network performs the task of delivering the packet to multiple destinations. For fixed host networks, multicast is achieved by constructing a delivering tree. Multicast applications such as weather reports, travel information and stock market reports may become widely used by mobile users. Multicast addresses are defined independent of location and separate from the normal unicast addresses. According to this characteristic, mobility should not be a problem for multicast.

The current multicast protocols in the Internet, Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), Protocol Independent Multicast-Sparse Mode (PIM-SM), Protocol Independent Multicast-Dense Mode (PIM-DM), and Core Based Tree (CBT) implicitly assume static hosts when building a multicast delivery tree. The challenging problem is how to achieve multicast service for MNs on the move. For a MN that wishes to receive a group multicast packet, it should join the group. The delivery trees with the MNs as receivers are actually dynamic delivery trees. The IETF Mobile IP working group proposed the Mobile IP to support unicast IP routing for mobile nodes in an IP internetwork [10]. Mobile IPv4 proposes two approaches to support mobile multicast. They are remote subscription and bi-directional tunneling.

One of the design goals of the Internet Protocol version 6 (IPv6) was to include multicast as part of the base standard, not as an add-on. Instead of having a separate group membership discovery, the IPv6 equivalent of Internet Group Management Protocol (IGMP) will be a part of Internet Control Message Protocol version 6 (ICMPv6) and will be present in all IPv6 nodes and routers. The set of ICMPv6 messages that enable Multicast Listener Discovery (MLD) [5] are the multicast listener “query,” “report,” and “done,” which are roughly equivalent to IGMP membership query, report and leave. There is no group owner at the IP level and joining the group is only a routing issue.

The contribution of this paper is a proposed scheme that implements hash algorithm to construct the multicast delivery tree of the MN movement and update the tree when the MN changes its point of attachment, participates in multicast session or leaves the multicast group. Therefore, unless the multicast group membership changes, no tree maintenance is required

In mobile IPv6, each IPv6 mobile node has at least two addresses per interface, namely the home address which is an IP address that is permanent to the mobile node and the care–of address, which is associated with the mobile node when it visits a particular foreign subnet [9].

Each time the mobile node moves from one subnet to another, it gets a new care-of address by stateless or stateful address autoconfiguration, such as Dynamic Host Configuration Protocol for IPv6 (DHCIPv6) [2].

It then registers it's Binding (association between a mobile node's home address and its care-of address) with a router in its home subnet, requesting this router to act as the home agent for the mobile node. This router registers this binding in its Binding Cache. At this point, the router serves as a proxy for the mobile node until the mobile node's binding entry expires. The router intercepts any packets addressed to the mobile node's home address and tunnels them to the mobile's care-of address using IPv6 encapsulation. The MN sends also a binding update to its Correspondent Nodes (CN), which can then learn and cache the new mobile's care-of address. As a result of this mechanism, when sending a packet to any IPv6 destination, a host must first check if it has a binding for this destination. If a cache is found, the host sends the packets directly to the care-of address indicated in the binding, using an IPv6 Routing header. If no binding is found, the packet is sent to the MN's home address, which tunnels it to the care-of address as described previously. When sending a packet to a CN, an MN may use its home address as source address.

Several architectures have been proposed to provide IP mobility support in IPv4. Balakrisnan *et al.* [1] proposed that packets destined for a mobile node are delivered to the mobile node's home agent using the IETF Mobile IPv4 and are then multicast to multiple base stations in close vicinity of the mobile node. In fact, packets destined for a MN have to transit through the home agent that can be distant from the mobile node's current location. This has the effect of increasing packet delivery latency, handoff latency and the Internet load. In the proposed scheme, the User Mobility Agent (UMA), which has the functionality as the home agent [1], is located in vicinity of the mobile node. In Mobile IPv6 binding updates are sent from the MN to the CN with every move [9]. Although this alleviates the triangle routing problem in Mobile IP, the communication overhead is still high during handoff rendering Mobile IP unsuitable for micro mobility and causing it to be unsuitable for audio applications.

Caceres *et al.* [3] proposed a hierarchical mobility scheme that separates three cases: local mobility, mobility within an administrative domain and global mobility. This hierarchical scheme is performed by extending Mobile IPv4 to include a hierarchy of foreign agents. In this proposal, each subnet that a MN could visit has one or more subnet foreign agents, which manage local mobility. On top of those subnet foreign agents, a domain foreign agent manages mobility across the different subnets of an administrative domain. The MN's home agent only keeps track of the movement of the MN across administrative domain boundaries. As a result, the MN's motion within an administrative domain is transparent to the home agent and its CNs. The hierarchical architecture of this scheme is similar to Castellussia scheme in [4] with the exception that the latter uses two levels of hierarchies: domain and global mobility. Caceres's scheme strongly relies on the deployment of foreign agents, which may not be available on a short term basis. Actually, Mobile IPv6 specification does not use foreign agents anymore. In contrast, our proposed scheme relies on IP multicast, which is already widely available.

An end-to-end architecture is proposed for IP mobility, based on dynamics Domain Name System (DNS) updates [13]. Whenever the MN moves, it obtains a new IP-address and updates the DNS mapping for its host name. A migration process is required to maintain the connection. The transport protocol is aware of the mobility mode during the migration process. Such architecture avoids triangle routing. However, such architecture incurs similar handoff delays to those experienced in Mobile IPv6 [6], or even worse due to DNS update delays and migration delays. The end-to-end approach is geared toward TCP-based applications, but it is not suitable for real-time multimedia applications (such as audio).

An approach for providing Mobility Support using Multicast (MSM-IP) is presented by Mysore and Bharghavan [8]. In their approach, each MN is assigned a unique multicast address. Packets sent to the MN are destined to that multicast address and flow down the multicast distribution tree to the MN. This is similar, in concept, to the Daedalus project approach [12]. However, it is not the Home Agent (HA) that tunnels the packets using the multicast address, rather, it is the CN that sends packets directly to the multicast address. This approach avoids triangle routing in addition to reducing handoff latency and packet loss by potentially using advance buffering. In MSM-IP, a hierarchy of servers is proposed for location management. Such hierarchy is complex, susceptible to failures and imposes restrictions of placement of the Rendezvous Point in PIM-SM as an underlying multicast protocol. In addition, by using binding updates and using the destination option in IPv6, the MSM-IP problems with TCP and other protocols can be avoided due to use of multicast addresses for MN. The multicast address is used within the network for packet routing, but the applications are only aware of the permanent unicast home address of the MN. Hence, no change to the application or transport protocol is needed.

Authors in [11] provide a comprehensive overview of existing multicast solutions to handle mobile sources and receivers in mobile IPv4 and mobile IPv6.

In this paper, a multicast-based scheme is presented for supporting IP mobility. Triangle routing, caching drawbacks suggested by the mobile IP architectures and creating our own hierarchy of agents have been avoided. Rather, a mechanism to reduce join and leave latencies have been proposed by implementing hash algorithm.

2. Proposed Architecture of the Hash Families

The sender may not be able to maintain very large groups (of thousands of members) by itself. Therefore, a division of the multicast group into subgroups (hash families) is proposed, sub-grouping scales well to very large group sizes. Each subgroup may contain some members. The initiator of these subgroups may be a third party entity such as Internet Service Provider (ISP) routers, User Mobility Agent (UMA) or other multicast routers. When a new host wants to join a multicast group, it selects the subgroup or the hash family. This depends on the value that was calculated by the initiator to the hash function.

The hash function is an algorithm that takes key field value as input and produces the relative address value within admissible range as output. The best hash function is the simplest one that sufficiently minimizes collisions (collision is the phenomenon when two distinct key values hash to the same location). There are many hash functions, but the best is based on modulo arithmetic and careful selection of table size. If the table size is prime, an even distribution of objects in the table will be obtained [7].

In the simulation model that has been built, the division-remainder (modulus) method was implemented based on the above characteristics and the following equation was used to calculate the hash function:

$$H(K) = K \bmod N \tag{1}$$

Where:
H (K): Hash key in the range [1, N].
K: Host identity.
N: Size of the hash table.

To improve the quality of the hash function, prime numbers were used for the routing tables, such as: 101, 211, 307, 1007, 10007, and 49999.

The multicast group is divided into several subgroups; each subgroup contains members that have the same value of the hash key (hash to the same entry in the routing table). The members of the subgroup were arranged within the table in a chain structure (hash family). Each chain starts at the hash key calculated in equation (1), and stored in the hash table. The hash table is stored in the initiator, UMA or any agent. A *'clue'* has been added to link the members of each subgroup through the chain, this will decrease the time delay to search the members of the subgroup when joining or leaving the multicast group.

3. Proposed Dynamic Addition and Deletion of Group Members

Multicast group members are added and deleted as follows:

- Every time there is a change in the group membership, the initiator calculates the hash key using equation (1) and modifies the group to either include or exclude a participant from further communication. The initiator assists in hash key calculation. It also assists in the building of hash families from the multicast group members. The hash families are stored in the routing table.
- The initiator divides the multicast group into several subgroups arranged in chains or hash families. Each chain starts at the hash key that is calculated using equation (1). The members of each hash family is linked to each other by a link address '*clue'*. The clue refers to the location of other members in the hash family within the routing table.
- The routing table is a random access table. Any host willing to join or leave the multicast group, the initiator first generates a key for the new host based on its identity; it then searches for an empty location in the chain to add the new host. In case of deletion, the initiator, after calculating the key, deletes membership of the leaving host. In both cases, the initiator recalculates the hash key and updates the hash family. Algorithm pseudo codes for both joining and leaving process are as follows:

Algorithm for a Host Joining a Multicast Group

```
Begin
    Calculate the hash key of the host using its identity
    IF the location in the hash table is empty Then
        Add the host to the multicast Group.
        Add clue to link host location.
        Update the hash family
    Else
        Search for an empty location
        Add the host to the group
        Update the hash family
End
```

Algorithm for a Host Leaving a Multicast Group

```
Begin
    Calculate the hash key of the leaving host using its
    identity
    Search the hash family starting from the hash key
    of the leaving host using clues
    IF the host is found THEN
        Delete its membership
        Update the hash family
    Else
        Play an error message, such as
        communication error and exit
End
```

4. Simulation Model

In order to provide mobility, the packets sent to the MN need to be forwarded to every location visited by the MN. Forwarding takes place according to the pattern of movement. The set of locations that the mobile node will visit may be viewed as a group of receivers, to which the packets should be delivered. The pattern by which packets are delivered to these receivers represents the component of the movement.

Instead of sending their packets to a unicast address, nodes wishing to send to the MN send their packets to a multicast group address. The MN through its movement would join this multicast group through the locations it visits.

In order to study the performance of the proposed scheme, a simulation model has been build for the construction of the multicast delivery tree. The simulation model adds a new branch to the tree when a member joins the multicast group and prunes the branch when a member leaves the multicast group. The simulation model consists of:

1. A model for initializing the random access routing table with different sizes (101, 211, 307, 1007, 10007, and 49999) to be used for building the hash families from the members of the multicast group.
2. A model for a host willing to join the multicast group. It calculates the hash key for the joined host. Also, it adds the new member to the suitable hash family such that, each family consists of members with the same hash key. The family members are connected to each other through a chain within the routing table. A link address '*clue'* is added to facilitate the search for any member in the group, hence reduce the time delay, which will result in smooth handoff.
3. A model for a host that wishes to leave the multicast group. In order to reach the member in its own group, the model recalculates the hash key, follows the chain using the link address then excludes the member from further communications, after updating the chain and the link address.

The above three models build the multicast delivery tree according to the MN movement. The statistics of the cumulative time to join/leave the multicast group by the group members are calculated.

Another model was added to convert the identity of the node from hexadecimal (i. e., in IPv6) to a form that is suitable for the implementation of the hash algorithm.

5. Results and Discussion

To illustrate the performance of the proposed algorithm, different sizes for the routing table have been used, with different number of members joining and leaving the multicast group. The cumulative time and average time for each member have been calculated in both joining and leaving cases. The cumulative time presents the time of joining or leaving the multicast group by a group of members, while the average time is the mean time taken by each member to join or leave the group.

Figures 1 and 2 illustrate the cumulative time that has been calculated by implementing the simulation model for joining/leaving the multicast group with sizes 1007 and 10007, respectively.

As shown in Figure 1 and 2, the leave time is much smaller than the join time. This difference is due to the fact that leaving the group means searching for the member through already constructed tree only and then deleting its membership. While joining the multicast group means adding a new branch to the multicast delivery tree. This addition includes searching for an empty space in the routing table besides updating the value of the link address in accordance.

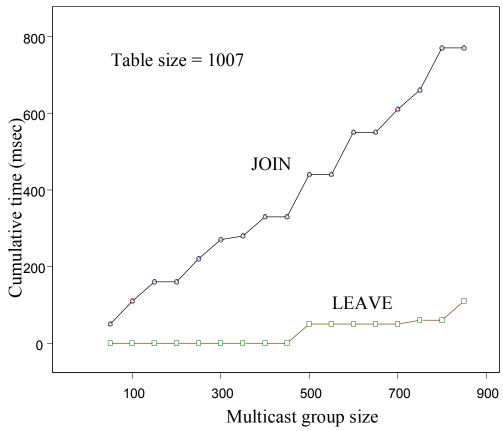

Figure 1. Cumulative time to join/leave the multicast group (table size = 1007).

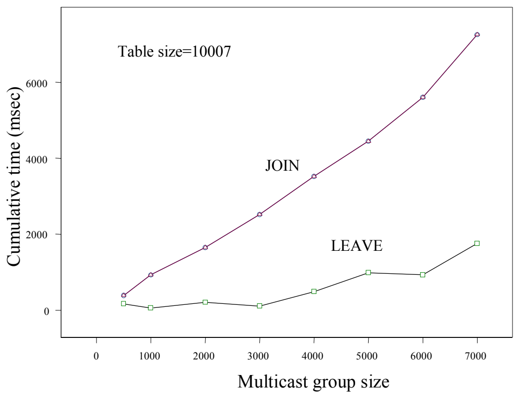

Figure 2. Cumulative time to join/leave the multicast group (table size = 1007).

The difference in cumulative time between the process of joining and leaving the multicast group is clearly shown in Table 1 for a hash table size of 1007. Notice that the real time clock that records the join/leave time jumps from 0 msec to 50 msec. This is due to the fact that Pentium III 900MHS 128MB RAM, simulation machine does not record any real time measurements below 50 msec.

Table 1. Cumulative time taken by hosts to join and leave a multicast group with hash table size = 1007.

No of Hosts	Join (ms)	Leave (ms)
50	50	0
100	110	0
150	160	0
200	160	0
250	220	0
300	270	0
350	280	0
400	330	0
450	330	0
500	440	50
550	440	50
600	550	50
650	550	50
700	610	50
750	660	60
800	770	60
850	770	110

Figure 3, 4, and 5 show the average time delay taken by each member to join or leave the multicast group for different sizes of the hash table: 1007, 10007 and 49999 respectively.
The average time calculated includes:

- The time to calculate the hash key from the node identity for each member.
- The time for adding the member to the suitable subgroup according to the hash key. In other words, adding a branch to the multicast delivery tree when a member joins the group and pruning the branch when leaving.
- The time to update the value of the link address.

Figure 3 shows that the average time for joining a group by each member (hash table size = 1007) is between 0.8 and 0.9 msec. While the average time to leave the same group is always below 0.2 msec.

Figure 4 shows that increasing the multicast group member number to ten folds does not affect the average join or leave time of the multicast group. But, if the size of the hash table has been increased to the value of 49999 and therefore the number of members also increased, the average join and leave time delays start to increase accordingly, as shown in Figure 5. Although the average leave time increases steadily as the number of the multicast group members increases, it is always below 1 msec. On the contrary, the average join time seems to increase at a faster rate but never exceeds 5 msec. This is obviously due to the increase in collisions in the hash addresses assigned and the time needed to resolve them.

Figure 6 summarizes the value of the average time to join or leave the multicast group for different sizes of the hash table. The average leave time for each member is almost zero seconds for any table size below 307. It is also shown from the table that the average join time is high at the beginning of the simulation experiment and when the table size was very small. This is probably due to the initialization process of the table and delivery tree construction processes. As the table size increases, the average join time becomes low and stable for a wide range of table sizes.

However, when the table size exceeds 10 k members, the average join time increases rapidly indicating that the chains became too long due to collisions and it takes a longer time to search for a member.

Nevertheless, due to the nature of transient movement of the MN, there will be a dynamic equilibrium of joining/leaving the multicast group that reduces the average join/leave time and therefore improves the performance of the system, by giving more opportunity for the MN to find an empty space in the hash table to build a branch in the delivery tree.

To decrease the handoff latency, the proposed algorithm searches a random access routing table. This means that the search for any member starts at the value of the hash key and within the subgroup only. Also, the use of the link address helps in the search process and to speed up the update of the subgroup after any change.

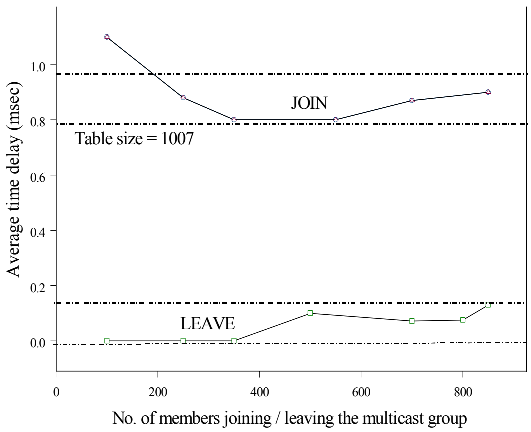

Figure 3. Average time delay for hash table size = 1007.

6. Conclusions

The actual design of mobile IPv6 is still a work in progress. The area of mobile IP multicasting is relatively new and there is not a widely accepted method for multicasting in such environment. The framework that has been proposed seems to provide a good basis for considering multicasting in mobile IPv6 networks. The architecture is multicast-based, in which a mobile node is assigned a multicast address, and the correspondent nodes send packets to that multicast group. As the mobile node moves to a new location, it joins the multicast group through the new location and prunes through the old location. Dynamics of the multicast tree provide for smooth handoff, efficient routing, and conservation of network bandwidth.

Security is always an issue for mobility support, where the continuous movement and change of attachment point is part of the normal operation. In the proposed scheme, however, the MN is assigned a location-independent multicast address that does not change with movement and hence does not reveal information about the MN's current point of attachment.

At any point in time, the MN should accept packets from only one location. However, during transient movement, the MN may be joined to the multicast group through multiple locations. The dynamics of joining and leaving/pruning the multicast group directly affect handoff latency and smoothness. To allow a smooth handoff, the MN should not prune the old location unless it starts getting packets from the new location. To further ensure smoothness and to conserve radio frequency bandwidth, the proposed algorithm was designed to reduce join and leave latencies.

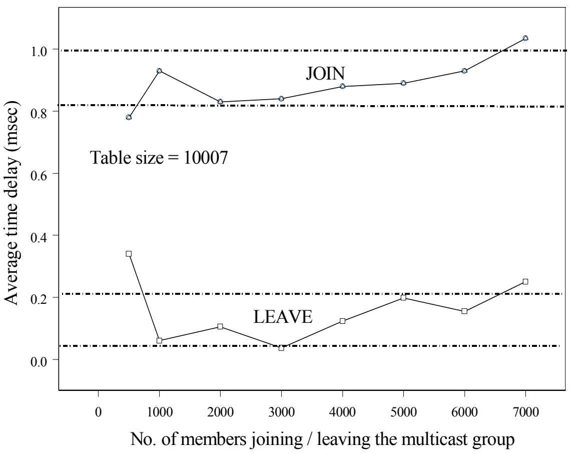

Figure 4. Average time delay for hash table size = 10007.

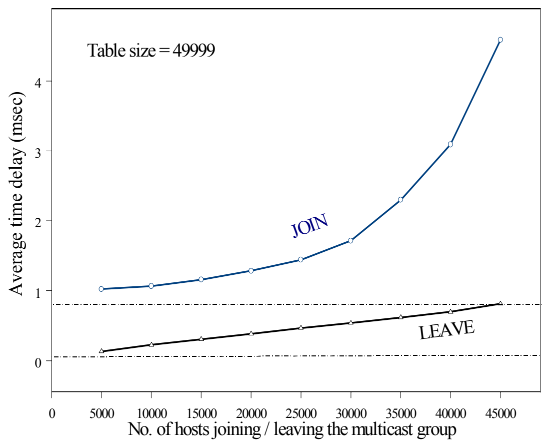

Figure 5. Average time delay for hash table size = 49999.

However, the study did not consider in detail the effect of collisions on the overall performance of the proposed handover algorithm due to limitations in space.

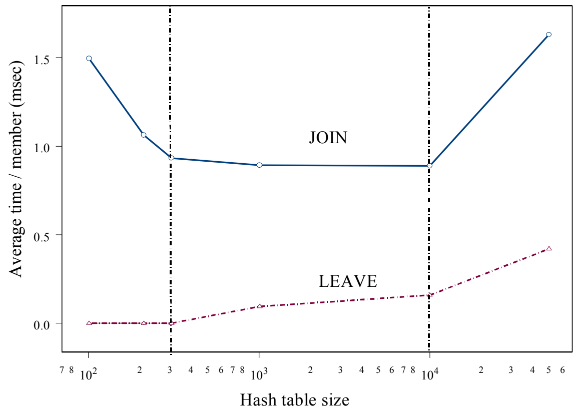

Figure 6. Average time to join/leave a multicast group for different sizes of the hash table.

Acknowledgements

This project is conducted as part of the Malaysian Intensified Research in Prioritised Areas (IRPA) 04-02-04-0002 project.

References

[1] Balakrisnan H., Seshan S., and Katz R., "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *in Proceedings of MOBICOM'95*, Berkeley, California, USA, November 1995.

[2] Bound J., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," *Draft-ietf-dhc-dhcpv6-19.txt*, 2001.

[3] Caceres R. and Padmanabhan V., "Fast and Scalable Handoffs for Wireless Internetworks," *in Proceedings MOBICOM'96*, Rye, New York, USA, November 1996.

[4] Castellussia C., "A Hierarchical Mobility Management Scheme for IPv6," *in Proceedings 3rd IEEE Symposium on Computers and Communications (ISCC'98)*, Athens, Greece, 1998.

[5] Deering S., Fenner W., and Haberman B., "Multicast Listener Discovery (MLD) for IPv6," *RFC2710*, 1999.

[6] Helmy A., "A Multicast-Based Protocol for IP Mobility Support," *ACM SIGCOMM, 2nd International Workshop on Networked Group Communication (NGC'2000)*, Palo Alto, 2000.

[7] Knuth D., *Art of Computer Programming: Sorting and Searching*, Addison-Wesley, 1998.

[8] Mysore J. and V. Bharghavan, "A New Multicasting-Based Architecture for Internet Host Mobility," *in Proceedings of ACM MobiCom*, Budapest, Hungary, Sptember 1997 .

[9] Perkins C. E. and Johnson D. B., "Mobility Support in IPv6," *in Proceedings of MOBICOM'96*, New York, USA, November 1996.

[10] Perkins C., "IP Mobility Support," *RFC2002*, 1996.

[11] Romdhani I., Khellil M., Lach H., Bouabdallah A., and Bettahar H.," IP Mobile Multicast: Challenges and Solutions," *IEEE Communications Surveys & Tutorials*, Q1, vol. 6, no. 1, pp. 18-40, 2004.

[12] Seshan S., Balakrishnan H. and Katz R., "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience," *Kluwer Journal on Wireless Personal Communications*, vol. 4, no. 2, pp. 141-162, Jan.1997.

[13] Snoeren A. and H. Balakrishnan, "An End-to-End Approach to Host Mobility," *ACM MobiCom*, Boston, Massachusetts, USA, August 2000.

Sahar Al-Talib received her BSc in engineering degree from University of Mosul, Iraq in 1976 and her MSc in engineering from University Putra Malaysia in 2003. Currently, she is working toward the PhD in the Department of Computer and Communication Systems, University Putra Malaysia. She is a student member in IEEE. Her research interest includes network engineering, algorithms, routing, multicasting, and IPv6.

Borhanuddin Mohd Ali received his BSc (Hons) in electrical and electronics engineering from Loughborough University of Technology in 1979, his MSc, and PhD in electromagnetics engineering from the University of Wales (Cardiff), in 1981 and 1985, respectively. He became a lecturer at the Department of Electronics and Computer Engineering, University Putra Malaysia in 1985, associate professor in 1993, and professor in 2003. Currently, he is the director of the Institute of Multimedia and Software, within the same university. He served short stints at Celcoms R&D in 1995 and Might in 1997. In 1996, he helped to realize the formation of Teman project, and more recently has been appointed as the protem chairman of the MyREN research community. He is a chartered engineer and a member of the IEE, and senior member of IEEE. He has been the chair of IEEE Malaysia Section 2002-2004, and previously the chair of ComSoc Chapter, 1999-2002. His research interest includes spans wireless and broadband communications, and network engineering, including IPv6.

Sabira Khatun received her BSc (Hons), MSc in applied mathematics, and PhD on hydromagnetic stability from the University of Rajshahi, Bangladesh in 1988, 1990, and 1994, respectively. She received her second PhD in communications and networking from University Putra Malaysia in 2003. She became a lecturer at the Discipline of Computer Science and Engineering, University Khulna, Bangladesh in 1991 and assistant professor in 1994. She joined as senior lecturer at the Department of Computer & Communication Systems Engineering, University Putra Malaysia in 1998. She is an active researcher of Teman project and MyREN Research Community. She is a member of IEEE. Her research interest includes spans broadband and wireless communications, and network management, including software defined radio and IPv6.