Mobile IPv6 Fast Handover Techniques

Cong Hung. TRAN [a], Van T.T DUONG [b]
[a] Post & Telecommunication Institute of Technology, Viet Nam
[b] Ton Duc Thang University, Viet Nam
conghung@ptithcm.edu.vn, vanduongthuy@yahoo.com
Addr: 11 Nguyen Dinh Chieu street, Dist.1 , Ho Chi Minh city, Fax: +84839 105510, Tel: +84903 918043

Abstract – **In the current mobile network context, a Mobile Node can change its point of attachment to different networks so frequently that basic Mobile IPv6 has many difficulties in preversing system quality in handover process. This problem has attracted the attention of many research groups and also gained certain achievements. Our paper focuses on analysing, comparing and synthesizing published Mobile IPv6 fast handover techniques. Based on these approaches,we propose an innovative method applying knowledge base on Mobile IPv6 Handover Problem to improve the handover performance of Mobile IPv6 by reduce the latency as far as possible.**

Our paper is organized as follows: Section 1 is the introduction, Section 2 presents Mobile IPv6 fast handover techniques which have been proposed. Section 3 presents our proposed solution. Section 4 presents performance evaluation and we conclude this paper in section 5.

Keywords – **Mobile IPv6, Fast Handover, Handoff, IP mobility, MIPv6, FMIPv6.**

1. INTRODUCTION

The Mobile IPv6 specification enables hosts to change their point of attachment to the Internet while not breaking existing application sessions. The process that a Mobile Node (MN) changes its point of attachment to the network or moves from one network to another network is known as handoff or handover. In the next mobile network technologies, when the MN usually changes its point of attachment to the Internet, the handover performance needs to be improved, especially the latency. Until now, there are many approaches to solve the fast handover problem. In the technique using RA (Router Advertisement) Caching to optimize movement detection of MN, there are research works such as [2], [3]. Other author teams have proposed that the movement detection, CoA (Care-of Address) configuration, and DAD (Duplicated Address Detection) processes are additionally executed in the AR (Access Router) instead of the MN as in the basic Mobile IPv6[4]. Meanwhile, [5] suggests cooperation between Layer 2 and Layer 3 handovers to decrease the handover delay.

Our paper provides readers with literature survey of proposed Mobile IPv6 extension techniques so that they can propose the models supporting fast handover.

2. MOBILE IPV6 FAST HANDOVER TECHNIQUES

2.1 *The basic Mobile IPv6 Handover*

Access Routers periodically multicast unsolicited RA message. When the MN receives RA, it compares the IP address in RA with its HoA (Home Address) to detect that whether it has moved to a new network. The two continuous RA messages take the time (RA period) between the minimum RA Interval (*30ms*) and the maximum RA Interval (*70ms*). To reduce the movement detection time, the MN can send RS (Router Solicitation) message to request AR reply RA for it. However, there must be a data link layer trigger in this case. The MN configures new CoA using information in RA.

Next, the MN needs to verify the uniqueness of its new CoA on the new link through DAD procedure. In order to perform DAD, the MN has to send one or several NS (Neighbor Solicitation) to its new CoA and wait for a response for at least one second. *This produces important additional time to handover latency*.

Once the new CoA construction is done, the MN must update the binding cache in its HA and CN by sending a Binding Update (BU) message.

2.2 *Mobile IPv6 fast handover Techniques*

Now, there are various techniques to extend the basic Mobile IPv6 handover. In general, there are two ways to extend the handover mechanism, based on entities in the network infrastructure and based on MN:

- The first way is to make modifications or add extensions to the entities in the network infrastructure such as routers, base stations. Routers or base stations can be changed so that they will only send RA to the MN when a handover is necessary as

- opposed to periodically sending RAs as the basic Mobile IPv6. This means the approximate location and signal strength of the MN need to be cached in nearby routers or base stations.
- The second way is to make modifications or add extensions on the MN. In this case, it is the MN that decides when a handover is appropriate.

The following are typical research works which have been proposed:

The technique using RA caching in AP (Access Point) to help MN for fast Access Router discovery [2] reduces movement detection time of MN. In this technique, the AP is added two functions, that are "*RA Caching*" and "*AP Notification*".

"*RA Caching*" caches RA message beforehand and sends it to a MN as soon as L2 association is made. Concretely, the AP scans all incoming L2 frames. It scans L2 frame header to see whether it is a multicast frame. If not, the AP sends that frame down link and scans next L2 frame. If so, the AP looks IP header to check whether it contains unsolicited RA from a AR. If incoming L2 frame doesn't contain unsolicited RA, the AP sends that frame down link and scans next L2 frame. When the AP finds unsolicited RA, it stores that frame and sends a copy down link.

When a MN moves to a new AP, it sends Association Request Message with its MAC address. Then, the AP grants association by sending Association Response Message. At that time, MN associates with the AP. As soon as association is made, the AP sends stored RA to the MN with MAC address in Association Request message. In this technique, MN receives RA just after association is made which is the earliest possible time in current standard.

The second technique using RA cache in the handover module on MN [3] provides the MN with the capability of filtering RAs to avoid the default processing of handover. The MN receives and processes cached RAs immediately after it moves to a new network and thus movement detection time is reduced. In addition, the priority of the RAs stored in the cache enables the MN to determine the best AR to connect. Four criteria are used to determine the priority of the RAs stored in the cache (sorted by the importance):

- The link signal strength (signal quality)
- The time since the RA entry was last updated
- The number of hops to the AR on the foreign network
- whether or not the AR is link local (priority is given to link local ARs since this may help the MN to detect that it is still reachable via its current IP address).

Unlike above techniques, the fast handover technique which based on AR [4] has proposed that the L3 movement detection, CoA configuration, and DAD processes are additionally executed in the AR instead of the MN as in the basic Mobile IPv6. After the completion of L2 handover, the MN and nAR may know it by some L2 triggers. After the nAR receives L2 triggers, it compares the L2 identifier of a MN (MAC address in case of 802.11) with the values in neighbor cache that contains the information about nodes connected to oneself:

- The L2 identifier is not found in neighbor cache, which means that the MN is a newcomer of the subnet area of the nAR. So the nAR prepares CoA generation and DAD operation.
- The L2 identifier is found in neighbor cache, which means that the MN is already served by the nAR. So the nAR does not have to do CoA generation and DAD operation. The only thing needed is that the nAR must immediately inform it to the MN and enables the MN to continuously use the existing CoA. This situation is that the MN is moved between different Access Points but APs are connected to the same AR. So, L2 handover happens, but L2 handover does not happen.

If the AR decides to generate a CoA for a MN, it uses its prefix information and L2 information of the MN which is included in L2 triggers. After configuration of a new CoA, the AR does DAD operation. When the MN requests a CoA through a modified RS message, the nAR responds to the MN with a modified RA message which includes the new CoA. On the other hand, the nAR can send the modified RA message immediately after the completion of DAD, even though it didn't receive RS from MN. At that time, the nAR does the normal neighbor discovery protocol to acquire an IP address of the MN. Then, the nAR can deliver the new CoA to the MN without a solicited message.

Similar to [4], the technique using cooperation between Layer 2 and Layer 3 handover [5] has proposed to execute movement detection, CoA configuration and DAD processes in the AR instead of in the MN. The overall handover procedure is presented by figure 1. When nAP receives the layer 2 reassociation.request from the MN, it provides the MN's MAC address to the nAR. Then, the nAR compares the MAC address to the information in its cache to identify whether the MN is a newcomer. If the MN is a newcomer, the nAR will generate a new CoA using the MN's MAC address and the prefix of the nAR and then do DAD process. These procedures can be finished before the nAR receives the RS message from the MN. So, the nAR can send immediately RA message included nCoA to MN.

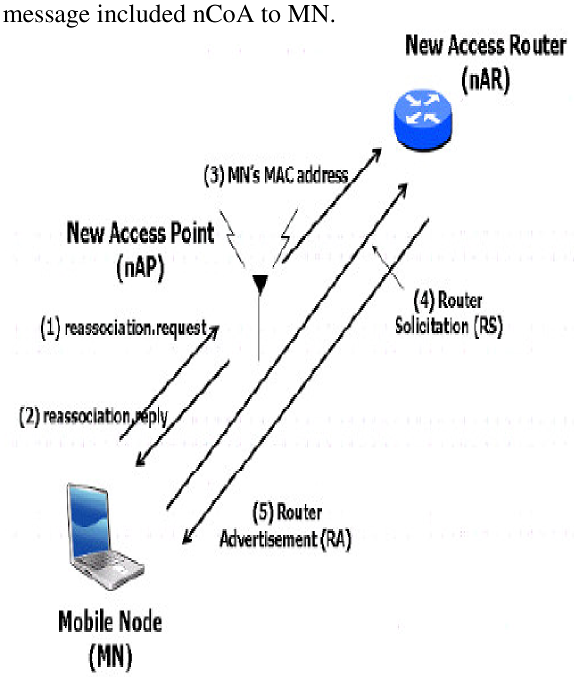

Figure 1. Handover procedure

In this technique, the movement detection, CoA configuration, and DAD processes are additionally executed in the nAR during the layer 2 handover, so the L3 handover delay is decreased considerably despite a little increase of the L2 handover delay. The experiment show that the total average handover delays are more than 1 second in Mobile IPv6 but 417.9ms in this mechanism.

3. PROPOSED SOLUTION

In the 802.11 network, the previous handover techniques are not effective in situations MN usually moves to new networks. Concretely, when MN often moves to an AR, instead of re-configuring CoA and performing DAD, CoA can be stored and reused in the following times. The DAD operation is the most time-consuming stage in the handover process so if it can be skipped, the latency will be greatly reduced. Experiments show that the total average latency of basic Mobile Ipv6 is 1295.2 ms while the average latency of DAD is 1000 ms [5].

In this paper, we propose a handover technique in order to solve that problem. The main idea is similar to the technique in [3], based on MN to extend Mobile IPv6 handover mechanism. We suggest supplementing an additional procedure AR Caching to MN for storing all the Access Routers that it connected in the past. Every record in AR Caching contains information of AR which it connected to, CoA address, connection instant, connection interval. When MN moves, it will perform data mining in AR Caching to predict the direction of movement. The predicted direction is the one towards AR to which it connected the most, with the top priority is connection instant. At that time, MN sends RS message to the nAR where it predicted that it is coming. If MN receives RA message from nAR, MN will search the corresponding CoA in AR Caching instead of re-configurating CoA based on RA as in basic Mobile IPv6. Consequently, in this case, CoA and DAD configuration procedures are skipped so the latency is reduced significantly.

3.1 Mobility Prediction

In fact human activities are often iterative. For example, a human being goes from home to his office in the weekday morning and vice versa in the weekday afternoon. At the weekend, he often goes to park. Therefore, a MN usually moves through a certain number of routers and the movement direction of MN depends heavily on travel time. As a result, we can use the movement history of MN to predict its movement direction. In this mechanism, we propose to add a cache to MN to store the information about all of its connections in the past, known as AR Caching. When MN moves, it will query AR Caching to determine AR towhich it often connected based on time travel. The query result is a list of ARs arranged in order of the number of times that the MN connected to. These are ARs where MN predicted that it is coming. At that time, MN sends RS message to ARs in the list in order of priority. The AR accepts the request of MN by sending RA message to MN. Thus, MN can discovery AR during its movement, so movement detection delay is reduced.

Consider a database, D, containing a log of a MN's mobility history. A mobility history is created whenever MN performs a handover to a new access router (nAR). The ID of this nAR is appended to the history (D). The IDs in two consecutive rows in D are neighbouring cells in a network. The first step requires the generation of mobility rules from database D. Each rule r: c_i - >c_j is coupled with a count value (count), which is the number of times that MN has moved from access router c_i to access router c_j.

In our method, we use a directed graph G, where the cells in the coverage region are considered to be the vertices of G. The edges of G are formed as follows: If two cells, say A and B, are neighboring cells in the coverage region then G has a directed and unweighted edge from A to B and also from B to A. The edges demonstrate the fact that a node can move from A to B or B to A directly. In Fig 2, an example coverage region and the corresponding graph G is presented.

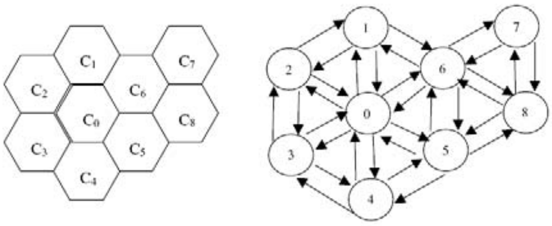

Figure 2. An example coverage region and the corresponding graph G[13]

An example database is given in Table 1. The last record in the database indentifies the MN's current position on the network, say *c*.

Table 1. Database of user mobility

Database D	Mobility rules	Count
<1>	1→6	2
<6>	6→5	2
<5>	6→1	1
<8>	5→8	1
<6>	5→0	1
<5>	8→6	1
<0>	0→6	1
<6>		
<1>		
<6>		
…		
<c>	$r_i \to r_j$	$>min_{count}$

Database D is MN's mobility history. The IDs in two consecutive rows in D are neighbouring cells in a network. Suppose that the ID in row i is c_i and ID in row i+1 is c_j, then the MN has moved from c_i to c_j.

The algorithm we have developed for user mobility rules mining is presented in Fig.3.

Mobility Rules Mining()

Input: Database D, number of access routers, n
Coverage region graph, G
Minimum value for count, min_{count}

Output: Set of mobility rules, R

```
R=∅
k=0
while(k<n)
{
    for each a_i ∈ D and a_i.ID=k
        if R contain rule r: k→a_{i+1}.ID
            r.count = r.count + 1
        else //Add the new rule into the set
  //of mobility rules, R
        {
            R = R ∪ r: k→a_{i+1}.ID
            r.count = 1
        }
    k=k+1
}
return R
```

Figure 3. Mobility Rules mining algorithm

By using the mobility rules mining algorithm, all possible mobility rules are generated and their count values are calculated. Then the rules which have a count higher than a predefined count threshold (min_{count}) are selected. These rules are used in the next phase of our algorithm, which is the mobility prediction.

Mobility Prediction()

Input: Current AR of the user, c
Set of mobility rules, R
Maximum predictions made each time, m

Output: Set of predicted cells, PCells

```
PCells = ∅
k=0
for each rule r: c_i→c_j ∈ R //check all the rules in
//R find the set of matching rules
{
    if c_i = c then //Add the rule into the set of
//matching rules
    {
        MatchingRules = MatchingRules∪r
//Add the (c_j, r.count) tuple to the Tuples array
        TuplesArray[k]= (c_j, r.count)
        k=k+1
    }
}
//Now sort the Tuples array w.r.t the "count" in
//descending order
TuplesArray←sort(TuplesArray)
index=0
//Select the first m elements of the Tuple Array
while (index < m && index < TupleArray.length
{
    PCells = PCells∪TupleArray[index]
    index = index +1
}
return PCells
```

Figure 4. *Mobility Prediction algorithm*

In the mobility prediction phase, the next movement of the MN is predicted. The mobility prediction procedure is presented in Fig.4. In this algorithm, we define a parameter, m, which is the maximum number of predictions that can be made each time the user moves. For prediction, we select the first m tuples from the sorted tuples array. Then the access routers of these tuples are our predictions for the next movement of the mobile node.

3.2 CoA configuration and Duplicate Address Detection

When MN receives RA message from nAR, the MN compares prefix information with the values in "AR Caching". The comparison results in two cases.

1) Prefix information is found in "AR Caching". It means that the mobile node has ever connected to this new access router. So the mobile node can reuse CoA of corresponding connection in "AR Caching" instead of re-configurating CoA based on RA as in basic Mobile IPv6. Consequently, in this case, CoA and DAD configuration procedures are skipped so the latency is reduced significantly.
2) Prefix information is not found in "AR Caching". It means that the mobile node is a newcomer of the subnet of this new access router. So the mobile node must configure a new CoA and do DAD operation following normal Mobile IPv6 mechanism.

4. PERFORMANCE EVALUATION

To evaluate the delay performance of the proposed scheme and compare to the Mobile IPv6 scheme, we used ns-2 in Ubuntu 8.9. The network topology in the simulation is shown by figure 5. There are two ARs and three APs. We assume that the MN starts from AP1, goes to the direction of AP3, and then returns to AP1. And we also assume that the MN usually moved between the two ARs in the past. We calculate the handover delays every time the MN moves between the ARs.

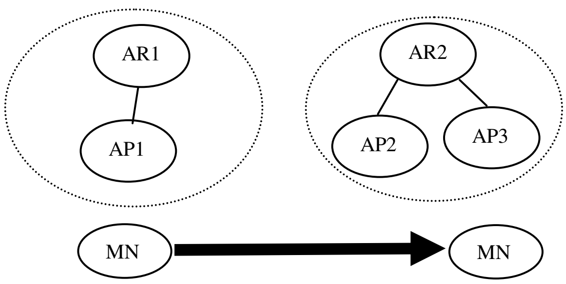

Figure 5. Simulation network topology

In this simulation, we can see that the handover delay is decreased independently of the period of RA messages from access routers.

5. CONCLUSION

In summary, Mobile IPv6 handover is an important and indispensable problem in future mobile network technologies. The largest issue in this problem is to minimize the collapse of system quality in Mobile IPv6 handover process. In particular, the issue that satisfies real-time handover has attracted many research efforts. This paper has presented different Mobile IPv6 fast handover techniques that research groups have proposed in previous reports. Currently, we continue to research reducing the latency to improve Mobile IPv6 fast handover problem as in proposed solution.

In future, we will conduct further analysis of the performance evaluation using NS-2 in Ubuntu. And we will compare the performance to the other schemes.

REFERENCES

[1] Tran Cong Hung, L. Phuc , N. N. Duy, D. N. T. Quynh, B. K. Jung, Y. Kang, "*Mobility Management in Wireless Mesh Networks*", ETRI-PTIT Collaborated Research Project, 2009.
[2] J .Choi, and D. Shin, "*Fast Router Discovery with RA Caching in AP*", IETF, Internet Draft draft-jinchoi-dna-frd-00.txt, 2004, expired.
[3] L. Patanapongpibul, G. Mapp, "*A client-based handoff mechanism for mobile IPv6 wireless networks*", ISCC, 2003.
[4] Y. Hong, M. Shin, H. Kim, *Access Router Based Fast Handover for Mobile IPv6*, IEEE Xplore, 2004.
[5] Seung Wook Moon and Jong Hyup Lee, *Reducing Handover Delay in Mobile IPv6 by cooperating with Layer 2 and Layer 3 Handovers*, ICACT, 2008
[6] D. Johnson, C. Perkins, and J.Arkko, "*Mobility Support in IPv6*", IETF RFC 3775, June 2004, http://www.ietf.org/rfc/rfc3775.txt.
[7] R. Pries, A. M'ader, D. Staehle, and M. Wiesen, "*On the Performance of Mobile IP in Wireless LAN Environments*", SpringerLink, University of W'urzburg, Institute of Computer Science, Germany, 2007.
[8] J. Kempf, M. Khalil, and B. Pentland, "*IPv6 Fast Router Advertisement*", IETF Internet Draft, draft-mkhalil-ipv6-fastra-05.txt, 2004, expired.
[9] Hsieh, R. Seneviratne, A. Soliman, H. El-Malki, K., *Performance analysis on hierarchical Mobile IPv6 with fast-handoff over end-to-end TCP*, Global Telecommunications Conference, page(s): 2488- 2492 vol.3, GLOBECOM '02. IEEE, 2002.
[10] Sulander, M.; Hamalainen, T.; Viinikainen, A.; Puttonen, J., *Flow-based Fast Handover method for Mobile IPv6 network*, Vehicular Technology Conference, 2004. VTC 2004-Spring, Page(s): 2447 - 2451 Vol.5, IEEE 59th, 2004.
[11] Sharma, S.; Ningning Zhu; Tzi-cker Chiueh, *Low-latency mobile IP handoff for infrastructure-mode wireless LANs*, Selected Areas in Communications, IEEE Journal on Volume 22, Page(s): 643 – 652, Issue 4, 2004.
[12] R. Chakravorty, P. Vidales, L. Patanapongpibul, K. Subramanian, I. Pratt and J. Crowcroft, *On Inter-network Handover Performance using Mobile IPv6*, University of Cambridge Computer Laboratory Technical Report, June 2003.
[13] G.Yavas, D. Katsaros et al., *A data mining approach for location prediction in mobile environments*, Data & Knowledge Engineering 54 (2005), page 121-146.
[14] Andre E. Bergh and Neco Ventura, *Prediction Assisted Fast Handovers for Mobile IPv6*, 2006.
[15] Rajeev *S.* Koodli and Charles E. Perkins, *Mobile inter-networking with IPv6: concepts, principles, and practices*, John Wiley & Sons, Inc., New Jersey., 2007.
[16] N. Montavont and T. Noël, *Handover Management for Mobile Nodes in IPv6 Networks*, IEEE Communications Magazine, August 2002.
[17] *The Network Simulator ns-2 manual*, at website http://www.isi.edu/nsnam/