Journal of Computer Sciences 2 (1): 01-06, 2006
ISSN 1549-3636

Home Agent Architecture and Algorithms for Mobility Management in Mobile IP Networks

Marcellin Diha and Samuel Pierre
Mobile Computing and Networking Research Laboratory (LARIM)
Department of Computer Engineering, Ecole Polytechnique of Montreal, P.O. Box 6079
Station Centre-ville, Montréal, Que., H3C 3A7, Canada

Abstract: This study proposes new Home Agent architecture and algorithms for real-time mobility management in mobile IP networks. The proposed architecture and algorithms offer a better performance based on the call-to-mobility ratio (CMR) and require less time for the location update and the tunneling compared with the Mobile IP model. These results are very useful and interesting for a real-time context where the factor time is very important.

Key words: Home agent architecture, call-to-mobility ratio, proposed algorithms, time factor

INTRODUCTION

The development of cellular networks brought many changes in the telephony area. Voice and mainly data transmission have increased a lot. The traditional telephony made place to new types of services that integrate both voice and data. It leads to the design and implementation of new hybrid networks able to fulfill this need. The mean-term goal is to have cellular networks entirely based on the IP protocol. But several problems must be solved before achieving this goal. Among these problems is the mobility management in IP networks because originally these networks were designed fix. In addition, the real-time aspect that is not supported in the actual IP protocol must also be addressed.

The IETF developed a mobile version of IP protocol able to manage users' mobility in IP networks[1-3]. Three main algorithms are proposed for mobility management: users' registration to a local router, foreign routers' discovery and location update when users are away from their home network and finally tunneling and data routing to mobile users. The Mobile IP protocol is implemented in several local area networks and works well. But real-time features are not supported in the protocol. There are two major components in mobility management: handover management and location management[4-9].

Handover management is the way a network uses to maintain connection to a mobile user as it moves and changes its access point to the network. In general, there are two types of handover: intra-cell handover and inter-cell handover[4]. The first type occurs when within a cell a user experiences degradation of signal strength. This leads to a choice of new channels having better signal strength at the same Base Transceiver Station (BTS). The second type occurs when a user moves from one cell to another cell. In this case, the user's connection information is transferred from the old BTS to the new one. In both intra-cell and inter-cell handover, the following procedure is performed. First, the user initiates a handover procedure. Then the network or the mobile (depending on the unit that controls the handover operation) provides necessary information and performs routing operations for the handover. Finally, all subsequent calls to the user are transferred from the old connection to the new one.

Location management is the process a network uses to find t he current attachment point of a mobile user for call delivery. The first step of the procedure is the *location registration*. In this phase, the mobile user periodically notifies the network of its new access point. The notifications allow the network to authenticate the user and update its location profile. The second step is the *call delivery*. When a call belonging to a user reached the network, a search for the user's profile is made usually in a local database. Then the call is forwarded to the user based on the information contained in its profile.

Mobility support in the IP protocol has been developed by the IETF leading to the Mobile IP protocol[3,10]. Currently two versions of Mobile IP are available, versions 4 (IPv4) and 6 (IPv6). In this study we focus on IPv4 since it is actually the most implemented one.

A Mobile Node (MN) is a node able to move from one subnet to another without any need of changing its IP address. The MN accesses the Internet via a Home

Corresponding Author: Samuel Pierre, Department of Computer Engineering, Ecole Polytechnique of Montreal
P.O. Box 6079, Station Centre-ville, Montréal, Que., H3C 3A7, Canada
Tel.: (514) 340-4711 ext. 4685, Fax: (514) 340-5159

Agent (HA) or a Foreign Agent (FA). The Correspondent Node (CN) is a node establishing a connection with the MN. The HA is a local router on the MN's home network and the FA is a router on the visited network. The following operations are introduced by the Mobile IP protocol[2,3].

Discovery: How an MN finds an agent (HA or FA).
Registration: How an MN registers with its HA.
Routing and Tunneling: How an MN receives datagrams when visiting a foreign network[11].

Location management operations include agent discovery, movement detection, forming care-of-address and location update. Handover operations include routing and tunneling. Figure 1 illustrates Mobile IP network architecture.

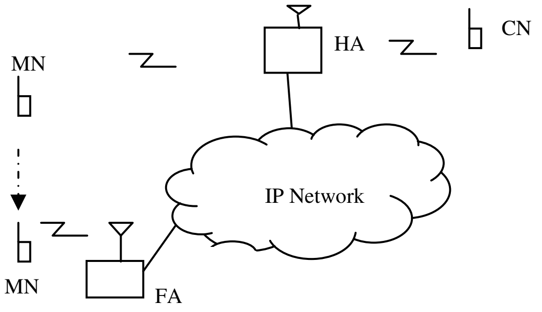

Fig. 1: Mobile IP network architecture

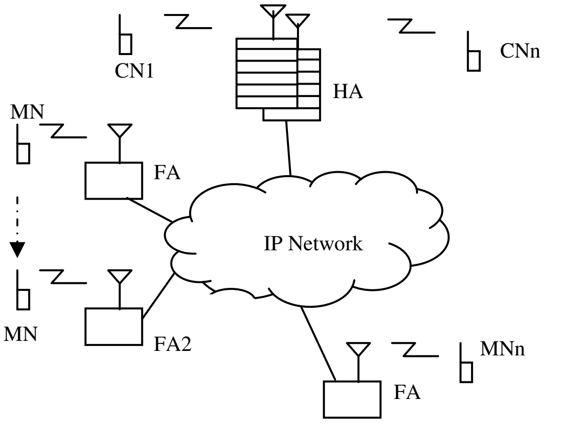

Fig. 2: Proposed architecture of Mobile IP network

Proposed architecture: Figure 2 shows the proposed Mobile IP network architecture. The architecture introduces the following main features:

* Real-time algorithms support.
* Connection of MNs and CNs to an FA or HA with different arrival rates in the network.
* All procedures associated with an MN (registration, discovery, tunneling and routing) represent different tasks with a specific priority.
* Multiprocessor agent (HA or FA). In this study the emphasis is put on the HA. Also the Home Agent is redundant to allow failure recovery.
* A main processor dispatching the different tasks arriving on an agent.
* A set of faster processors is defined to process high-priority tasks.
* Architecture allowing different speeds for the processors.

Proposed algorithms: Based on the architecture described above, a set of new algorithms has been defined for mobility management. These algorithms are derived from Mobile IPv4 algorithms[3]. They introduce the notion of priority management in a real-time context. The new discovery algorithm adds the ranging concept in addition to the lifetime used in Mobile IPv4. Also it allows the MN to initiate Foreign Agents search at startup in stead of waiting advertisements. In addition, this algorithm allows the MN to keep a list of the most recent Foreign Agents that it tries to contact first before initiating any broadcast search.

Tasks scheduling and assignment algorithm: The task scheduling and assignment algorithm is described in Fig. 3. The scheduling part of this algorithm is based on the EDF algorithm[12] while the assignment part is a totally new concept since it based on a multiprocessor architecture. The tasks are sorted based on the deadline and assigned to the processors. If a task is critical (short deadline), it is assigned to a faster processor. If not it is assigned to a normal processor. A task is assigned to a processor only if its current utilization rate is less than 1. This ensures that a processor is not used at its full capacity while others are unused.

```
-Given $i$ counter of tasks and $j$ counter of processors.
-Given $n$ tasks $TSK_1, \dots, TSK_i, \dots, TSK_n$ with the priorities
$p_1, \dots p_i, \dots p_n$.
-Given $S$ a set of faster processors and $p_s$ threshold of a critical task.
-Given $u(i)$ the utilization rate of the task $TSK_i$ and $U(j)$ a
vector of $u(i)$ on $P_j$.
BEGIN
  -$i = 1, U(j) = 0, S = 1$.
  -Sort the $n$ tasks based on $u(i)$ on $P_0$.
  WHILE $i \leq n$ DO
      $j = \min\{k | U(k) + u(i) \leq 1\}$
      IF $( (p_i > p_s) \&\& (\exists P_R \in S \mid \sum U(P_R) < 1)) )$
THEN $TSK_i \longrightarrow P_R$
      ELSE $TSK_i \longrightarrow Pj$
      $i \longleftarrow i + 1$
  END WHILE
```

Fig. 2: Task scheduling and assignment algorithm

Registration algorithm: The *registration procedure* is a task running on the HA with the *highest priority*. It can preempt any other mobility management task for a given user. For example, during a tunneling procedure, if a registration request is received for the same user, the tunneling process will be delayed until the registration is done. The different stages of the algorithm are described as follow:

* MN sends a registration request to the HA.
* HA verifies IF a task other than the registration is in process for the same user.
IF yes THEN the task is preempted by the registration task.
* HA sends a response to the MN.
* IF request accepted THEN registration procedure done ELSE MN retries UNTIL request accepted.

Discovery algorithm: The discovery algorithm introduces also the notion of priority in a real-time environment and it is based on the lifetime expiration and the ranging. The *discovery procedure* has the *second highest priority*. The different steps are described as follow.

* IF first time startup THEN MN sends a broadcast advertisement.
* FAs verify if no higher priority task is being executed for the same MN.
IF yes THEN delay discovery process UNTIL high-priority task execution is done.
* FAs send responses back to MN.
* MN chooses FA with most strong signal strength and records the lifetime, the care-of-address and the FA's IP address.
* IF lifetime expires or the MN starts going out of range (wick signal strength) THEN send registration request to Foreign Agents in the MN's local database.
IF no FA responds back THEN broadcast a discovery advertisement message.
* REPEAT steps 2 through 4 UNTIL registration succeed.
* IF registration succeeds THEN MN sends new location information to HA for location update.

Routing and tunneling algorithm: The new routing and tunneling algorithm also introduces the notion of priority in a real-time environment. *This procedure* has the *lowest priority*. Thus, during a tunneling procedure, if a registration procedure is received for the same user, the location procedure will be suspended until the registration is done. The steps of the algorithm are the following:

* HA receives data for an MN.
* HA verifies if a registration request is made for the same user.
IF yes THEN HA suspends tunneling process until registration is done.
* IF MN in local network THEN delivered packets using normal IP packets delivery procedure ELSE forward packets to MN via its current FA.

Performance analysis: The performance analysis is based on the CMR (Call-to-Mobility Ratio). The CMR is the average number of messages send to a user divided by the average number of networks or subnets visited by the user in a given time stamp. The goal of the CMR analysis is to determine the ratio by which the proposed model reduces the location update and the tunneling times.

$$CMR = \frac{\lambda}{\mu} \qquad (1)$$

Where, λ is the average number of messages send to a user and μ the average number of subnets or networks visited by the user between two consecutives messages.

We define the following parameters to compare the CMR in the Mobile IP and the proposed models:

U cost for location update procedure execution in the Mobile IP model;
L cost for tunneling procedure execution in Mobile IP model;
u cost for location update procedure execution in the proposed model;
l cost for tunneling procedure execution in the proposed model;
T cost to cross a boundary between two subnets;
U_{MIP} total cost for the location update procedure in Mobile IP model;
L_{MIP} total cost for tunneling procedure execution in Mobile IP model;
C_{MIP} total cost for location update and tunneling procedures execution in Mobile IP model;
U_{PROP} total cost for location update procedure in the proposed model;
L_{PROP} total cost for tunneling procedure execution in the proposed model;
C_{PROP} total cost for location update and tunneling procedures execution in the proposed model.

The total costs are obtained as follows:

$$U_{MIP} = \frac{U}{CMR} \qquad (2)$$

$$C_{MIP} = U_{MIP} + L_{MIP} + T = \frac{U}{CMR} + L + T \qquad (3)$$

$$U_{PROP} = \frac{u}{CMR} \qquad (4)$$

$$C_{PROP} = U_{PROP} + L_{PROP} + T = \frac{u}{CMR} + l + T \qquad (5)$$

Our goal is to reduce the location update and the tunneling times by respectively at least 50% and 25%. Then *u* = *U/2* and *l* = *L/4*. We make the following assumptions to simplify the analysis *L* = *U*, *T* = *U/4* = *u/2* and *U* = *1*, that leads to:

$$\frac{U_{MIP}}{U_{PROP}} = \frac{U}{u} = \frac{1}{u} \qquad (6)$$

$$\frac{L_{MIP}}{L_{PROP}} = \frac{L}{l} = \frac{2}{u} \qquad (7)$$

$$\frac{C_{MIP}}{C_{PROP}} = \frac{U(4+5CMR)}{4u+2CMR} = \frac{4+5CMR}{4u+2CMR} \qquad (8)$$

Figure 4 shows the location update cost for different values of the CMR and *u* = *0.2*.

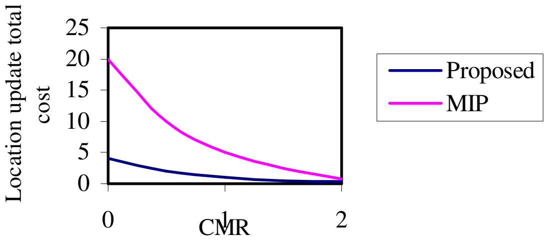

Fig. 3: Comparison of location update total cost (*u=0.2*)

Figure 5 illustrates the location update cost for different values of the CMR and *u* = *0.5*.

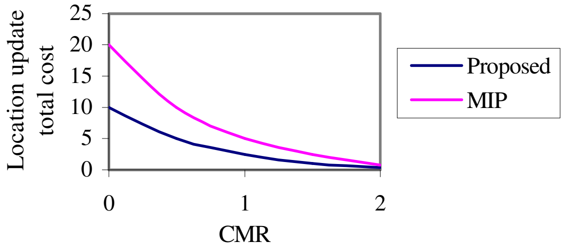

Fig. 5: Comparison of location update total cost (*u=0.5*)

Figure 6 shows the location update cost and the ratio C_{MIP}/C_{PROP} for different values of the CMR for *u* = *0.2* and *0.5*.

In Fig. 4 and 5 we noticed that the proposed model reduces by 80% the location update time for u = 0.2 and by 50% for u = 0.5 for small values of the CMR. The reason is that in this case the mobile users make an important number of location update requests and the processing on the multiprocessor HA in the proposed architecture is faster compared with the mobile IP model. When the CMR increases, the mobile users stay longer in the same network and in this case, the location update time decreases and is near 0 for the two models. For the total costs (location update plus tunneling) shown in Fig. 6, the reduction is between 67% and 80% for u = 0.2 and between 50% and 60% for u = 0.5.

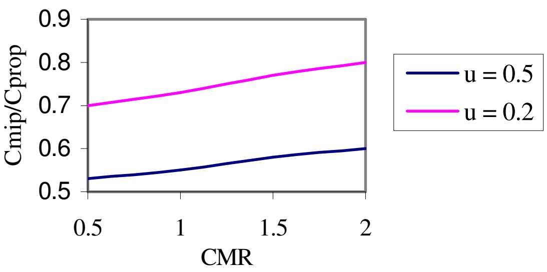

Fig. 4: Comparison of ratio C_{MIP}/C_{PROP}

We can conclude that the proposed model offers a better performance based on the CMR. Indeed, it takes less time for the location update and the tunneling in the proposed model compared with the Mobile IP model. These results are very useful and interesting for a real-time context where the factor time is very important. To validate the analysis, we had conducted different simulations.

COMPUTATIONAL RESULTS

Figure 7 shows the setup for the different simulations. The network used is an Ethernet based LAN 10/100 Mbps with an 8-Port Ethernet Hub.

We simulated the implementation of the current mobile IP algorithms as well as the proposed architecture and algorithms in a real-time environment using VxWorks as real-time Operating System running on a MPPC (Motorola^{TM} Power PC). Our simulations focus on the location update average time, the tunneling average time, the number of tasks missing their deadline depending on the number and the speed of the processors on the HA.

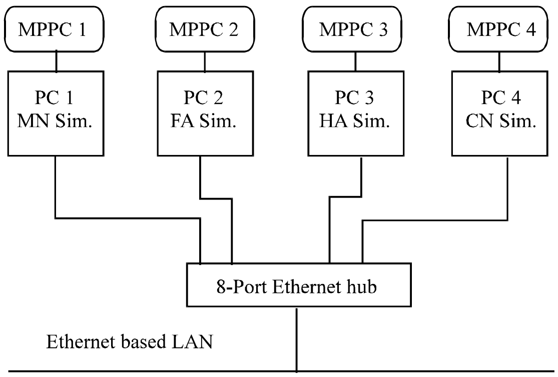

Fig. 5: Simulation setup

Location update average time: In the Mobile IP model, the location update time is constant and does not depend on the user's arrival rate in the network. The location update average rate value is around 1 sec. In the proposed model, the location update time increase with the user's arrival rate. The maximum value is around 0.8 s. For small value of the arrival rate, few users arrived in the network and are dispatched on the processors, which in this case have a low utilization rate. The processing is then faster per user leading to a low location update time. But when the arrival rate increase, many users arrived in the network and the processing takes longer but the average time is still lower compared with the Mobile IP model. Overall the location update time is reduces by 20 % to 80% in the proposed model, which is above the targeted objective of 50%.

Figure 8 shows the location update average time for different arrival time in the networks.

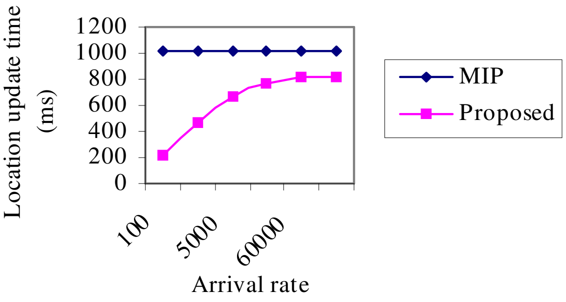

Fig. 6: Location update average time

Tunneling average time: Figure 9 illustrates the tunneling average time for different data sizes.

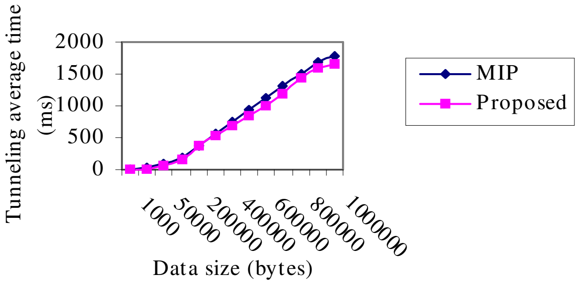

Fig. 7: Tunneling average time

The tunneling time is smaller in the proposed model compare to the Mobile IP model. The data processing is faster because of the multiprocessor architecture. The messages sent to the mobile users spend less time in the message queue on the HA. In the mobile IP model the message *i + 1* will wait longer than the message *i* on the message queue, increasing the processing time. But in the proposed model the message *i + 1* can be process in parallel on a different processor while processing the message *i* on an other one. As a result the total time spent in the system is reduced. Overall the tunneling time is reduces by 10 to 30% in the proposed model, which is above the targeted objective of 25%.

Task scheduling and assignment: Figure 10 shows the number of tasks missing their deadline for different number of processors with different speeds following Gaussian and exponential distributions.

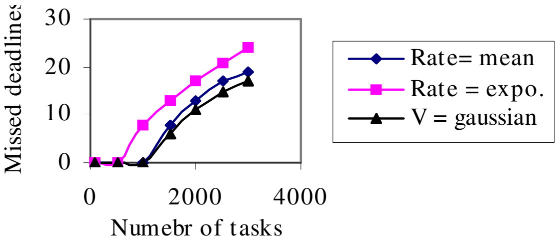

Fig. 10: Processors with different speeds distribution

The number of tasks missing their deadline in the Gaussian distribution is lower compare to the exponential distribution. The reason is that in the first case the speeds of the processors are close to the mean speed. It is the contrary in the exponential case where the distribution is larger with more low speeds. This leads to a higher ratio Execution Time/Processor Speed and number of missed deadlines. So, for configurations with different speeds, the speeds of the processors must follow a Gaussian distribution in order to have an optimal scheduling and assignment for the tasks.

CONCLUSION

In this study we presented a Mobile IP architecture and mobility management algorithms in a real-time context. The implementation of the proposed architecture and algorithms gave better results for the location update and tunneling average times as well as the CMR compare to the existing architecture and algorithms[13-14]. The location update time is reduced by 20 to 80% while the tunneling time is reduced by 10 to 30%. These results meet time constraint in real-time systems. The multiprocessor architecture is the core of the proposed model. It gives a faster parallel processing for the mobile users.

The scheduling and assignment algorithm is optimal for different number of processors with different speeds. This achievement is something new compare to actual real-time multiprocessor scheduling and assignment algorithms. In the current algorithms, the processors must have the same speed to guarantee an optimal scheduling and assignment.

Many investigations are on going in real-time mobility management for Mobile IP networks. The areas cover the implementation of real-time algorithms in real networks as well as proposition of new algorithms and architectures. Also, since the current protocols are designed for micro-mobility, the WAN and global roaming areas are some new domains of interest.

REFERENCES

1. 3GPP, 2000. Combined GSM and Mobile IP Mobility Handling in UMTS IP CN. 3GPP Technical Specification.
2. James, D.S, 1998. Mobile IP, The Internet Unplugged. Prentice Hall, PTR.
3. Perkins, C., 2002. IP Mobility Support for IPv4. Internet Engineering Task Force, RFC 3220.
4. Akyildiz, I.F., J. McNair, J.S.M. Ho, H. Uzunalioglu and W. Wang, 1999. Mobility Management in Next-Generation Wireless Systems. Proc. IEEE., 87: 1347-1384.
5. Stallings, W., 2001. Wireless Communications and Networks. Prentice Hall, Upper Saddle River, New Jersey.
6. Caceres, R. and V. Padmanabhan, 1996. Fast and scalable handoffs for wireless networks. In Proc. ACM/IEEE MOBICOM'96, pp: 56-66.
7. Wong, V.W.S. and V.C.M. et Leung, 2000. Location Management for Next Generation Personal Communication Networks", Department of Electrical and Computer Engineering. The University of British Columbia, Vancouver, Canada, Tech. Rep.
8. Stallings, W., 2001. Wireless Communications and Networks. Prentice Hall, Upper Saddle River, New Jersey.
9. Ho, J. and I.F. et Akyildiz, 1995. Mobile user Location Update and Paging under Delay Constraints. ACM/Baltzer. J. Wireless Networks, 1: 413-425.
10. Perkins, C., 1996. Minimal Encapsulation within IP. Internet Engineering Task Force, RFC 2004.
11. Calhoun, P. and C. Perkins, 1996. IP Encapsulation within IP. Internet Engineering Task Force, RFC 2003.
12. Chrishna, C.M. and G.S. Kang, 1997. Real-Time Systems. McGraw Hill.
13. Schneider, J. and R. et Comroe, 2001. Motorola Canopy TM Access Point User's Guide.
14. Schneider, J. and R. et Comroe, 2001. Motorola CanopyTM Subscriber Modem User's Guide.
15. Gupta, V., 1998. Solaris Mobile IP: Design and Implementation. SUN Microsystems Inc.