2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies

Distributed Route Selection Algorithm in Nested Multihomed Mobile Networks

Sulata Mitra and Sumanta Pyne, Department of Computer Science & Technology
Bengal Engineering and Science University, India
e-mail:mitra_sulata@hotmail.com

***Abstract*—The present work is a distributed route selection algorithm in nested multihomed mobile networks. The mobile network node sends a request message to all leaf mobile routers inside the mobile network to initiate a session. All leaf mobile routers execute route selection algorithm to select the best route for the desired session of the mobile network node and send the best route to the mobile network node. The mobile network node transmits the packets corresponding to the desired session using the best route to Internet. The mobile routers associated with the best route execute egress interface selection algorithm to select the best egress interface and deliver the packets corresponding to the desired session of the mobile network node using the best egress interface to the next hop of the selected route. The performance of the proposed work is evaluated on the basis of throughput, session loss and route selection time using NEMO_SIM simulator. Results based on a detailed performance evaluation study are also presented to demonstrate the efficacy of the proposed scheme.**

I. Introduction

In 4G scenario users expect to be connected to the Internet from "anywhere" at "anytime", in fixed wireless locations or while on the move, provided that any available access network can be accommodated. For doing so, mobile networks (MNs) may be multihomed i.e. having multiple points of attachment to the Internet. Moreover a user may have more than one mobile device, say a mobile phone, a laptop and a personal digital assistant (PDA). Each of these devices could likely have multiple network interfaces that enable them to interconnect with each other as well as with other networks. These devices moving with the user together constitute public access network (PAN) and are an example of a small scale mobile network. The access networks deployed on public transportation such as ships, trains, buses and aircrafts are examples of mobile networks at a larger scale. Support for multihoming in a network mobility environment is crucial since if a mobile router (MR) fails to maintain session continuity this would affect the session preservation of the entire network. The multi-homing support would enhance the load sharing and fault tolerant capabilities of mobile networks.

The existing node mobility arrangement protocols, like MIP protocols [1,2] can not support the network mobility as the mobility service should be provided transparently to every node inside the network. A network mobility (NEMO) basic support protocol has been proposed [3] to support this kind of network. The NEMO basic support protocol is an extension of MIPv6 [2]. In [4] Cho et al. proposed a home agent based (HA-based) dynamic load sharing mechanism for multihomed mobile networks. The registered neighbor mobile router-Home agent (MR-HA) tunnels and measured MR-HA tunnel latency is required to provide HA based solution. A dynamic neighbor MR authentication and registration mechanism using the Return Routerability procedure of MIPv6 is considered in this work. The proposed scheme measures tunnel latency using periodic binding update (BU)/binding acknowledgement (BACK) messages and the HAHA protocol [5]. The HA can share traffic load with the neighbor MR-HA tunnel depending upon the measured tunnel latency. In [6] Shima et al. proposed two operational experiments of network mobility. The first experiment is based on NEMO basic support in a real environment. The real environment was the WIDE 2005 autumn camp meeting [6]. At the meeting a wireless network was provided to the attendees. The MR of the proposed mobile network had two network interfaces, one was for external connectivity and the other was used to provide the mobile network. But the result of this experiment shows a serious service disruption problem during handover. The second network mobility experiment uses the WIDE 2006 spring meeting environment [6]. The multiple care of address (CoA) registration mechanism [7] is used in this experiment which helps to use multiple network interfaces concurrently. The MR was equipped with three network interfaces. It can connect to a new network before leaving an old network. The multiple CoA mechanism is useful for seamless handover of a mobile network and the mobile network is practically usable as a moving network.

The present work (Fig.1) considers (n,1,1) [8] configuration of MN. The proposed MN has 6 mobile routers (MRs), single home agent (HA) and single mobile network prefix (MNP). The number of egress interface of each MR is assumed as 4. There are 4 possible routes of transmission from MN to Internet as shown in TABLE-1. The present work considers three sets as X=(E1,E2,E3,E4)=(X1,X2,X3,X4), Y=(Delay, Unused bandwidth, Packet loss, Cost)=(Y1,Y2,Y3,Y4) and Z=(data, voice, video)=(Z1,Z2,Z3). The set X indicates 4 egress interface of each MR, Y indicates 4 parameter values to determine the status of each egress interface of a MR and Z indicates 3 different service types that are supported by the MN. The parameter value Y1 is in msec, Y2 is in kbps, Y3 is in % and Y4 is in unit. In the present work both the egress interface selection algorithm and route selection algorithm assume maximum bandwidth requirement for data application to achieve fast data transfer whereas the moderate bandwidth requirement is assumed for voice and video application. Moreover both the algorithm assume voice application as delay sensitive and video application as loss sensitive.

The main objective of the proposed scheme is to select the best route for the mobile network node (MNN) from MN to Internet for their desired service type using route selection algorithm. Each MR in a MN determines the current status of each of its egress interfaces using egress interface selection algorithm and delivers the packets to the next hop of the selected route using the best egress interface. It also determines its own status depending upon the status of its egress interfaces and sends its status to all MRs in the MN. Each leaf MR ($leaf_{MR}$) executes the route selection algorithm using the same status parameter values independently to select the best route for the desired service type of MNN. The execution of the route selection algorithm in such a distributed way helps to protect the MN from the failure of any leaf MR. So at any instant of time the best possible route is selected for the desired service type of MNN. A new simulator (NEMO_SIM) is proposed in the present work. This simulator is a software which takes a NEMO as input and produces performance measurement of NEMO as output. So it can consider any NEMO as input. It is used to study the performance of the proposed MN (Fig.1).

TABLE-1

Route	Path
r1	MNN->MR2->MR1->Internet
r2	MNN->MR4->MR3->Internet
r3	MNN->MR6->MR5->Internet
r4	MNN->MR4->MR3->MR1->Internet

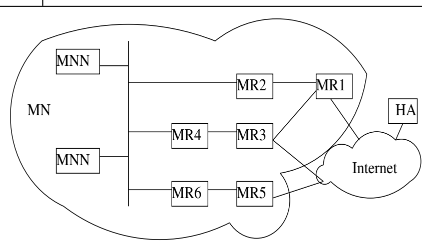

Fig. 1. Mobile network

II. Present Work

In this section the proposed scheme is considered for discussion.

2.1 Message Exchange among Various Nodes of MN: When a MNN wants to initiate a session, it sends MNN_leaf_{MR} message to all leaf MRs in the MN for the selection of a suitable route as source node. All leaf MRs store this message in a priority queue at its ingress interface and assign a priority value to each request of MNN. If the number of requests from a particular MNN increases the priority value assigns to each such request reduces. As a result when a new request arrives from a new MNN, the request is given a higher priority. Such consideration helps to prevent the flooding of the ingress queue at the ingress interface of all leaf MRs by the requests from malicious MNN. All leaf MRs remove a request from the queue after route selection. The MNN_leaf_{MR} message contains 2 component as MNN identification (MNN_id) and Service type (S_type). In case of 100000 MNN, the number of bits require to represent MNN_id is 17. In case of 3 different service types supported by the MN, the number of bits require to represent the service type is 2. So the length of this message is 19 bits. Each MR in the MN maintains the values of the parameters such as Delay, Unused bandwidth, Packet loss, Cost of its 4 egress interfaces in the form g(X,Y) as discussed in section 2.2.2 and also computes the value of its own status parameters as discussed in section 2.2.3. Each MR sends the status parameter values to all MRs in the MN independently if a change occurs in the value of status parameter(s) in the form of MR_MR message. After receiving this message each leaf MR computes the values of the parameters such as Delay, Unused bandwidth, Packet loss and Cost as discussed in section 2.3.1 of all the routes from MN to Internet. Each leaf MR also executes the route selection algorithm as discussed in section 2.3.2 to select the best route for the desired session of MNN after receiving MNN_leaf_{MR} message from MNN. It assigns an unique session identification (Session_id) to each session of MNN after selecting the best route for that session. Each leaf MR maintains one counter (session_count) to count the number of active session. The counter value increases by 1 after selecting each route per session. The number of bits require to represent Session_id is $log_2(session\_count)$. Each leaf MR also sends $leaf_{MR}$_MNN message to MNN after selecting the best route. This message contains the route identification (r_id) of the best route as selected by the route selection algorithm for the desired session of MNN and Session_id which is assigned to that session. In case of 4 routes the number of bits require to represent r_id is 2. So the length of this message is $2+log_2(session\_count)$. After receiving $leaf_{MR}\_MNN$ message, MNN initiates the session. The MNN specifies the value of Session_id, r_id, P_no (number of packets in the desired session), MNN_id and S_type in the header of the first packet whereas MNN_id and Session_id in the header of the remaining packets of the session. So the length of the header in the first packet is $21+log_2(session\_count)+log_2(P\_no)$ bits and the length of the header in the remaining packets is $17 + log_2(session\_count)$ bits.

2.2 Function of each MR: The function of each MR is discussed in this section.

2.2.1 Maintenance of Routing Table: Each MR maintains a routing table to keep the record of various sessions in the form (MNN_id, Session_id, P_no). The value of the attributes in each record are obtained from the header of the first packet corresponding to the session. One route is selected as the best route by the route selection algorithm for each session of a MNN and one record is maintained in the routing table for each such route. Each MR associated with the best route inserts a record in the routing table after receiving the first packet of that session. After transmitting each packet all the MRs associated with the best route reduce the value of P_no attribute in the corresponding record by 1. The leaf MR associated with the best route receives packet from MNN and the other MRs associated with the same route receive packet from their predecessor node. When a MR receives a packet, it searches the routing table using Session_id as the searching key to retrieve the corresponding record. If found, verifies the MNN_id and transmits the packet to the next hop of the best route provided the value of the P_no attribute in the corresponding record is nonzero. Each MR associated with the best route deletes a record from the routing table when its P_no attribute becomes zero. A route remains idle for a long time if the corresponding MNN becomes out of order or stops transmission or go out of the coverage area of MN. A route becomes out of order in case of failure of the link(s) associated with it. For such cases the MRs associated with the route delete the corresponding record from the routing table and make the resources associated with the route free which helps to improve the resource utilization of the MN.

2.2.2 Computation of g(X,Y): When a MNN sends a packet to the ingress interface of the leaf MR associated with the best route as selected by the route selection algorithm, it includes the current time stamp in the header of the packet. MR also measures the time stamp after transmitting the said packet to the next hop using its best egress interface as determined by the egress interface selection algorithm. The difference of the two time stamp (δt) is considered as the delay per packet for that MNN. The initial value of delay at j^{th} egress interface (Ej) ($Delay_{Ej}$) is assumed as 0.0 msec. Let δt_{ij} indicates the delay per packet for the service type of i^{th} MNN using Ej. So $Delay_{Ej}$ is increased by δt_{ij} after transmitting a single packet of i^{th} MNN.

In case the MRs are in the WiFi network, the available bandwidth per egress interface of the MR can be assumed as the bandwidth of WiFi network. The initial value of the unused bandwidth at Ej (un_BW_{Ej}) is assumed as the available bandwidth at Ej (av_BW_{Ej}) and $desire\_BW_{ij}$ indicates the bandwidth which is required for the service type of i^{th} MNN using Ej. So after receiving the first packet from i^{th} MNN, un_BW_{Ej} is reduced by $desire\_BW_{ij}$ and after receiving the last packet from i^{th} MNN, un_BW_{Ej} is increased by $desire\_BW_{ij}$. It is assumed that each MR knows the desired bandwidth for each of the 3 service type that are supported by MN.

The packet loss at any egress interface is the summation of the packet loss due to time out and buffer overflow. A counter is maintained at each egress interface to count the number of loss of packets. The initial value of packet loss counter at Ej (PL_{Ej}) is assumed as 0. Each MR searches all the packets in the buffer at Ej for time out and increases PL_{Ej} by 1 after removing a packet from the buffer at Ej due to time out. PL_{Ej} is also increased by 1 after removing a packet from the buffer at Ej due to buffer overflow. The packet loss at Ej is computed in % as (PL_{Ej}/total packet at Ej)$*100$.

The cost per egress interface is the summation of cost of all the MNNs using that particular egress interface. The cost of each MNN is the summation of route selection cost and transmission cost. The route selection cost depends upon the overhead due to message exchange for the selection of the route. Now the overhead due to message exchange is the summation of bits in MNN_leaf_{MR} message, $leaf_{MR}\_MNN$ message and the length of the header in all the packets from MNN. The transmission cost is the product of the amount of data in bits and cost/bit. Now the amount of data in bits is the product of the number of packet and size of packet (P_sz) in bits. The initial value of cost at Ej ($Cost_{Ej}$) is assumed as 0. Let $Cost_{ij}$ indicates the cost for the service type of i^{th} MNN using Ej, where $Cost_{ij} = [19 + (2 + log_2 session\_count) + (21 + log_2 session\_count + log_2 P\_no) + (P\_no - 1)(17 + log_2 session\_count) + (P\_sz * P\_no)]$*cost/bit. The present work assumes cost/bit as 1 unit. P_sz is assumed as 8000 bits, 640 bits and 712 bits for data, voice and video packet respectively. So after receiving the first packet of i^{th} MNN, $Cost_{Ej}$ is increased by $Cost_{ij}$. Each MR performs the same computation to calculate the 4 parameter values (Delay, Unused bandwidth, Packet loss, Cost) of all its 4 egress interfaces.

2.2.3 Computation of Status Parameter Values: The p^{th} MR (MR_p) computes $Delay_p$, un_BW_p, PL_p and $Cost_p$ as its own status parameters using the element values of gp(X,Y) (it is g(X,Y) as maintained by p^{th} MR). gp(X,Y) is defined as

$$\begin{bmatrix} Delay_{E1\_p} & un\_BW_{E1\_p} & PL_{E1\_p} & Cost_{E1\_p} \\ Delay_{E2\_p} & un\_BW_{E2\_p} & PL_{E2\_p} & Cost_{E2\_p} \\ Delay_{E3\_p} & un\_BW_{E3\_p} & PL_{E3\_p} & Cost_{E3\_p} \\ Delay_{E4\_p} & un\_BW_{E4\_p} & PL_{E4\_p} & Cost_{E4\_p} \end{bmatrix}$$

where $Delay_{Ej\_p}$, $un\_BW_{Ej\_p}$, PL_{Ej_p} and $Cost_{Ej\_p}$ are the Delay, Unused bandwidth, Packet loss and Cost of Ej ($1 \leq j \leq 4$) at p^{th} MR respectively. $Delay_p = (Delay_{E1\_p} \wedge Delay_{E2\_p} \wedge Delay_{E3\_p} \wedge Delay_{E4\_p})$;
$un\_BW_p = (un\_BW_{E1\_p} \vee un\_BW_{E2\_p} \vee un\_BW_{E3\_p} \vee un\_BW_{E4\_p})$;
$PL_p = (PL_{E1\_p} \wedge PL_{E2\_p} \wedge PL_{E3\_p} \wedge PL_{E4\_p})$;
$Cost_p = (Cost_{E1\_p} \wedge Cost_{E2\_p} \wedge Cost_{E3\_p} \wedge Cost_{E4\_p})$;

2.2.4 Egress Interface Selection Algorithm per Service Type: The leaf MR associated with the best route executes this algorithm after receiving the first packet from MNN. The other MRs associated with the best route execute this algorithm after receiving the first packet of MNN from its predecessor MR. This algorithm helps a MR to select the best egress interface for the desired service type of MNN as specified in the header of the first packet. The MR delivers the packet of MNN using its best egress interface to the next hop of the best route.

un_BW_p is computed in section 2.2.3. If $un\_BW_p = un\_BW_{Ej\_p}$, Ej is the best egress interface of MR_p for data service. $Delay_p$ is computed in section 2.2.3. If $Delay_p = Delay_{Ej\_p}$, Ej is the best egress interface of MR_p for voice service. PL_p is computed in section 2.2.3. If $PL_p = PL_{Ej\_p}$, Ej is the best egress interface of MR_p for video service.

2.3 Function of each leaf MR: Each leaf MR computes the values of the parameters such as Delay, Unused bandwidth, Packet loss and Cost of all the routes from MN to Internet after receiving MR_MR message. Each leaf MR also executes route selection algorithm after receiving MNN_leaf_{MR} message. The function of each leaf MR is considered for discussion in this section.

2.3.1 Computation of Parameter Values for 4 Routes: The MRs MR2 and MR1 are associated with the route r1. MR2 is the leaf MR associated with r1. The MRs MR4 and MR3 are associated with the route r2. MR4 is the leaf MR associated with r2. The MRs MR6 and MR5 are associated with the route r3. MR6 is the leaf MR associated with r3. The MRs MR4, MR3 and MR1 are associated with the route r4. MR4 is the leaf MR associated with r4. Each leaf MR computes the parameter values of all 4 routes from MN to Internet as follows:

Delay_r1, un_BW_r1, PL_r1 and Cost_r1 are the 4 parameter values corresponding to the route r1.

$Delay\_r1 = (Delay_{MR2} \vee Delay_{MR1})$

$un\_BW\_r1 = (un\_BW_{MR2} \wedge un\_BW_{MR1})$

$PL\_r1 = (PL_{MR2} \vee PL_{MR1})$

$Cost\_r1 = (Cost_{MR2} \vee Cost_{MR1})$

$Delay\_r2$, un_BW_r2, PL_r2 and $Cost\_r2$ are the 4 parameter values corresponding to the route r2.

$Delay\_r2 = (Delay_{MR4} \vee Delay_{MR3})$

$un\_BW\_r2 = (un\_BW_{MR4} \wedge un\_BW_{MR3})$

$PL\_r2 = (PL_{MR4} \vee PL_{MR3})$

$Cost\_r2 = (Cost_{MR4} \vee Cost_{MR3})$

$Delay\_r3$, un_BW_r3, PL_r3 and $Cost\_r3$ are the 4 parameter values corresponding to the route r3.

$Delay\_r3 = (Delay_{MR6} \vee Delay_{MR5})$

$un\_BW\_r3 = (un\_BW_{MR6} \wedge un\_BW_{MR5})$

$PL\_r3 = (PL_{MR6} \vee PL_{MR5})$

$Cost\_r3 = (Cost_{MR6} \vee Cost_{MR5})$

$Delay\_r4$, un_BW_r4, PL_r4 and $Cost\_r4$ are the 4 parameter values corresponding to the route r4.

$Delay\_r4 = (Delay_{MR4} \vee Delay_{MR3} \vee Delay_{MR1})$

$un\_BW\_r4 = (un\_BW_{MR4} \wedge un\_BW_{MR3} \wedge un\_BW_{MR1})$

$PL\_r4 = (PL_{MR4} \vee PL_{MR3} \vee PL_{MR1})$

$Cost\_r4 = (Cost_{MR4} \vee Cost_{MR3} \vee Cost_{MR1})$

2.3.2 Route Selection Algorithm: Each leaf MR in the MN executes this algorithm to select the best route for the desired service type of MNN independently after receiving MNN_leaf_{MR} message.

Route Selection for Data Service: The proposed algorithm selects the route having maximum unused bandwidth for data service. Each leaf MR computes a parameter value r_data as r_data=$(un\_BW\_r1 \vee un\_BW\_r2 \vee un\_BW\_r3 \vee un\_BW\_r4)$

If r_data=un_BW_r1, r1 is the best route for data service. Similarly if r_data=un_BW_r2 or r_data=un_BW_r3 or r_data=un_BW_r4, r2 or r3 or r4 is the best route for data service respectively.

Route Selection for Voice Service: The proposed algorithm selects the route having minimum delay for voice service. Each leaf MR computes a parameter value r_voice as r_voice=$(Delay\_r1 \wedge Delay\_r2 \wedge Delay\_r3 \wedge Delay\_r4)$. If r_voice=$Delay\_r1$, r1 is the best route for voice service. Similarly if r_voice=$Delay\_r2$ or r_voice=$Delay\_r3$ or r_voice=$Delay\_r4$, r2 or r3 or r4 is the best route for voice service respectively.

Route Selection for Video Service: The proposed algorithm selects the route having minimum packet loss for video service. Each leaf MR computes a parameter value r_video as r_video=$(PL\_r1 \wedge PL\_r2 \wedge PL\_r3 \wedge PL\_r4)$. If r_video=$PL\_r1$, r1 is the best route for video service. Similarly if r_video=PL_r2 or r_video=PL_r3 or r_video=PL_r4, r2 or r3 or r4 is the best route for video service respectively.

III. NEMO_SIM Simulator

The proposed work is simulated with the help of a NEMO_SIM simulator. It is an application based object oriented simulator. When an user gives a complete NEMO as input to NEMO_SIM, the NEMO_SIM automatically creates an environment of a NEMO where communication can take place. The NEMO_SIM is implemented using JAVA, because of platform free usage of the executable JAVA programs and also for further extension of the simulator to be accessed online. JAVA has a good set of Application Program Interfaces that largely benefits the development of complex simulation softwares. NEMO_SIM can be a part of NS2 simulation environment by using AgentJ [9], which is a JAVA Virtual Machine for NS2. NEMO_SIM can also act as an extended part of JNS 1.7, JAVA Network Simulator [10]. The NEMO in the proposed scheme is the combination of some interconnected processing units such as MNN, LFN, MR. Each processing units are treated as threads and the whole NEMO is considered as a complex producer-consumer problem in a large scale. JAVA provides facility of using multiple threads and thread synchronization which is the main ingredient for building NEMO_SIM. The function of all the threads are discussed in the following sections.

3.1 MNN_REQ Thread: It sends MNN_leaf_{MR} message to all leaf MRs.

3.2 $leaf_{MR}\_MNN$ **Thread:** It receives MNN_leaf_{MR} message request from MNN, runs the route selection algorithm and sends $leaf_{MR}\_MNN$ message to MNN.

3.3 MNN_SERVICE_START Thread: It receives $leaf_{MR}\_MNN$ response message and starts a new session.

3.4 MNN_SERVICE Thread: It creates a new session for the desired application, transmits packet corresponding to the desired application to the ingress interface of the leaf MR associated with the best route. After transmitting all the packets successfully this thread dies. A MNN has zero or more such thread depending upon how many sessions are still alive.

3.5 MR_STATUS_UPDATION Thread: It sends MR_MR message.

3.6 MR_MR Thread: It receives MR_MR message.

3.7 MR_PACKET_RECEIVE_FORWARD Thread: It receives a packet from the ingress queue and forwards it to the best egress as selected by the egress interface selection algorithm.

3.8 MR_egress Thread: It receives a packet from the best egress queue and forwards it to the ingress queue of the next hop. It also computes packet loss due to the overflow at the egress queue.

3.9 MR_EGRESS_PACKET_LOSS Thread: It discards the packets from the egress queue due to time out.

IV. Simulation

The simulation experiment is carried out considering the internal network of NEMO (Fig.1) as WiFi (IEEE 802.11a). The size of LFN buffer, MR egress as well as ingress buffer and MNN buffer are assumed as 1000, 10^5 and 1000 respectively.

Fig.2, Fig.3 and Fig.4 show the plot of throughput, session

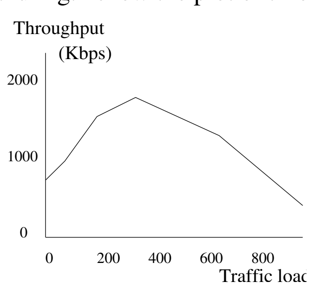

Fig.2 Throughput vs. Traffic load

loss and route selection time vs. traffic load of the proposed NEMO (Fig.1). The traffic load is computed as the ratio of arrival rate and departure rate of session request from MNN by all leaf MRs independently. The route selection time is constant up to traffic load 200 and then it increases slowly which causes decrease in throughput with traffic load. Session loss is zero up to traffic load 220 and then it increases slowly.

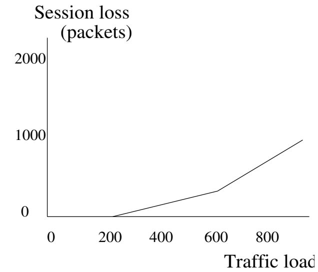

Fig.3 Session Loss vs. Traffic load

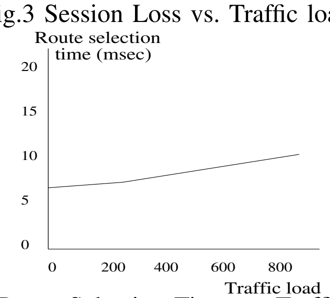

Fig.4 Route Selection Time vs. Traffic load

V. Conclusion

This paper has presented a distributed route selection algorithm in a multihomed mobile networks. The simulation result shows the efficiency of the proposed scheme in terms of throughput, session loss and route selection time. The proposed scheme can be extended to provide communication between MNN in MN and correspondent node (CN). In case of high network mobility communication between MNN and CN takes place through MR-HA tunnel whereas direct communication between MNN and CN is possible in case of lower network mobility to achieve route optimization.

References

[1] C.Perkins, ‘IP Mobility Support for IPv4’, IETF RFC 3344, August 2002.

[2] D.Johnson, C.Perkins, and J.Arkko, ‘Mobility Support in IPv6’, IETF RFC 3775, June 2004.

[3] V.Devarapalli, R.Wakikawa, A.Petrescu, and P.Thubert, ‘Network Mobility Basic Support Protocol’, IETF RFC 3963, January 2005.

[4] S.Cho, J.Na and C.Kim, ’A Dynamic Load Sharing Mechanism in Multihomed Mobile Networks’, pp.1459-1463, ICC 2005.

[5] R.Wakikawa, V.Devarapalli and P.Thubert, ’Inter Home Agents Protocol (HAHA)’, IETF Internet Draft, draft-wakikawa-mip6-nemo-haha-01, February 2004.

[6] K.Shima, Y.Uo, N.Ogashiwa and S.Uda, ’Operational Experiment of Seamless Handover of a Mobile Router using Multiple Care-of Address Registration’, Journal of Networks, vol.1, no.3, July 2006.

[7] R.Wakikawa, T.Ernst and K.Nagami, ’Multiple Care-of Adresses Registration’, IETF, Tech. Rep. draft-wakikawa-mobileip-multiplecoa-05, February 2006.

[8] T.Ernst and J.Charbon, ’Multihoming with NEMO Basic Support’,ICMU, Yokosuka, Japan, January 2004.

[9] I.Taylor, I.Downard, B.Adamson and J.Macker, ‘Agentj: Enabling Java NS-2 Simulations for Large Scale Distributed Multimedia Applications’, Second International Conference on Distributed Frameworks for Multimedia DFMA 2006, 1-7, Penang, Malaysia, May 2006.

[10] Java Network Simulator: http://jns.sourceforge.net