Security Issue of a Route Selection Algorithm in Multihomed Mobile Networks

Sulata Mitra and Sumanta Pyne
Department of Computer Science & Technology, Bengal Engineering and Science University, Shibpur
Howrah-711103, West Bengal, India
mitra_sulata@hotmail.com
sumantapyne@gmail.com

***Abstract*—The present work is a dynamic route selection algorithm in multihomed mobile networks. The mobile network node sends a request message to a local fixed node inside the mobile network to initiate a session. The local fixed node executes route selection algorithm to select the best route for the desired session of the mobile network node and delivers the packets corresponding to the desired session of the mobile network node 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 security issue of the proposed route selection algorithm and its solution is also considered in the present work. The performance of the proposed work is evaluated with and without incorporating the solution of the security issue using NEMO_SIM simulator. Results based on a detailed performance evaluation study are also presented to demonstrate the efficacy of the proposed scheme.**

Keywords-Multihomed Mobile Network, Egress Interface Selection Algorithm, Route Selection Algorithm, Multimedia Traffic

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. A local fixed node (LFN) inside MN uses dynamic route selection algorithm to select the best route for 3 different service types (data, voice, video) of mobile network node (MNN) independently. There are 4 possible routes of transmission from MN to Internet as shown in TABLE-1. The present work considers two sets as X=(E1,E2,E3,E4)=(X1,X2,X3,X4) and Y=(Delay, Unused bandwidth, Packet loss, Cost)=(Y1,Y2,Y3,Y4). The set X indicates 4 egress interfaces of each MR and Y indicates 4 parameter values to determine the status of each egress interface of a MR. The parameter value Y1 is in msec, Y2 is in kbps, Y3 is in % and Y4 is in unit.

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

The main objective of the proposed scheme is to select the best route for the MNN from MN to Internet for their desired service type. 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 LFN which helps LFN to select the best route for each service type of each MNN dynamically and independently. 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 assumes voice application as delay sensitive and video application as loss sensitive. So at any instant of time the best possible route is selected for the desired service type of MNN. The proposed scheme also considers the security issue of the route selection algorithm and its solution. A new simulator (NEMO_SIM) is proposed in the present work. This simulator is 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) after incorporating the solution for the security issue of the proposed route selection algorithm.

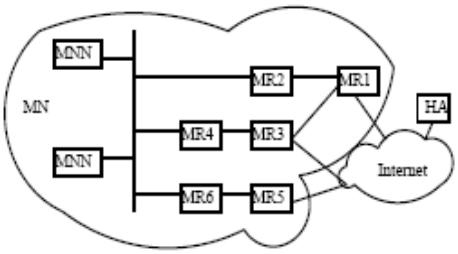

Fig. 1. Mobile network

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_LFN message as discussed in section 2.1.1 to LFN for the selection of a suitable route as source MNN. LFN stores this message in a priority queue and assigns a priority value to each request of MNN. LFN removes a request from the queue after route selection. In response to the MNN _LFN message LFN uses route selection algorithm to determine the best possible route from MN to Internet for the desired service type of the MNN and sends LFN_MNN message as discussed in section 2.1.2 to MNN. In response to the LFN_MNN message, MNN initiates the session and sends MNN_MR message as discussed in section 2.1.3 to the leaf MR associated with the best route. Each MR generates MR_ LFN message as discussed in section 2.1.4 after determining its own status parameters and sends it to LFN.

2.1.1 MNN_LFN message:

This message contains 2 components as MNN identification (MNN_id) which represents source identification of the session and service type (S _type). In case of 100000 MNN, the number of bits required to represent MNN_id is 17. In case of 3 different service types supported by the MN, the number of bits required to represent S_type is 2. So the length of this message is 19 bits.

2.1.2 LFN_MNN message:

It contains the identification of the selected route, the identification of the leaf MR associated with the selected route and session identification (Session_id). In case of 4 routes the number of bits required to represent the identification of the selected route is 2. In case of 6 MRs the number of bits required to represent the identification of the leaf MR is 3. Each session has a unique session identification number as assigned by LFN after selecting the best route for that session. LFN 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 $log_2(session\_count)$. So the length of LFN_MNN message is 5 + $log_2(session\_count)$ bits.

2.1.3 MNN_MR message:

The MNN_MR message has 3 different formats (Fig.2). The format as shown in Fig.2 (a) is used as the header of the first packet and its length is 26 + log_2S_no + log_2P_no + $log_2session\_count$ bits. The format as shown in Fig.2 (b) is used as the header of the last packet and its length is 2 + log_2S_no + $log_2session\_count$ bits. The format as shown in Fig.2(c) is used as the header for all the intermediate packets and its length is $log_2session\_count$ + log_2S_no bits. S_no indicates the sequence number of each packet. P_no indicates the number of packets in the corresponding session. S_flag indicates start flag. It is set in the

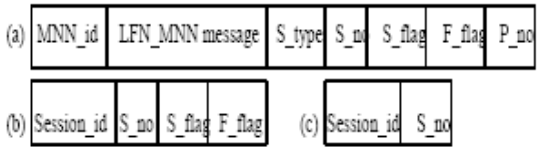

Fig. 2. MNN_MR message format (a) Packet header for first packet, (b) Packet header for last packet, (c) Packet header for intermediate packet

first packet of the session to indicate the start of the session. F_flag indicates finish flag. It is set in the last packet of the session to indicate the end of the session.

2.1.4 MR_LFN message:

Each MR maintains the values of the parameters such as Delay, Unused bandwidth, Packet loss and Cost of its 4 egress interfaces in the form g(X, Y) as discussed in section 2.2.2. Each MR also computes the value of its own status parameters as discussed in section 2.2.3 and sends it to LFN in the form of MR_LFN message provided a change occurs in the value of status parameter(s).

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, r_id). The value of the attributes in each record are obtained from the header of the first packet (Fig.2 (a)) 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. 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. Each MR associated with the best route inserts a record in the routing table after receiving the first packet of that session and deletes the record from the routing table after receiving the last packet of that session. 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 (r_id). 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 the 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 makes 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_LFN message, LFN_MNN message and MNN_MR message. 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 + (5 + $log_2 session\_count$) + (26 + $log_2 S\_no$ + $log_2 P\_no$ + $log_2 session\_count$) + ($log_2 session\_count$ + $log_2 S\_no$) + ((*P_no* - 2) * (2 + $log_2 S\_no$ + $log_2 session\_count$)) + (*P_no* * *P_sz*)]*cost/bit, In the present work cost/bit is assumed 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 ($1 \le j \le 4$) at p^{th} MR respectively. $Delay_p$ = ($Delay_{E1\_p}$ ^ $Delay_{E2\_p}$ ^ $Delay_{E3\_p}$ ^ $Delay_{E4\_p}$), un_BW_p = ($un\_BW_{E1\_p}$ ˇ $un\_BW_{E2\_p}$ ˇ $un\_BW_{E3\_p}$ ˇ $un\_BW_{E4\_p}$), PL_p = (PL_{E1_p} ^ PL_{E2_p} ^ PL_{E3_p} ^ PL_{E4_p}), $Cost_p$ = ($Cost_{E1\_p}$ ^ $Cost_{E2\_p}$ ^ $Cost_{E3\_p}$ ^ $Cost_{E4\_p}$).

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 MNN_MR message. 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 $un\_BW_p = un\_BW_{E\_p}$, 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 LFN:

LFN computes the values of the parameters such as Delay, Unused bandwidth, Packet loss and Cost of the 4 routes from MN to Internet after receiving MR_LFN message. LFN also executes route selection algorithm after receiving MNN_LFN message. The function of LFN is considered for discussion in this section.

2.3.1 Computation of parameter values for 4 routes:

After receiving MR_LFN message LFN computes 4 parameter values for each route from MN to Internet. The MRs MR2 and MR1 are associated with the route r1. Delay r1, un_BW_r1, PL_r1 and Cost_r1 are the 4 parameter values corresponding to the route r1.

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

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

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

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

The MRs MR4 and MR3 are associated with the route r2. *Delay_r*2, *un_BW_r*2, *PL_r*2 and *Cost_r*2 are the 4 parameter values corresponding to the route_r2.

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

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

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

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

The MRs MR6 and MR5 are associated with the route r3. *Delay_r*3, *un_BW_r*3, *PL_r*3 and *Cost_r*3 are the 4 parameter values corresponding to the route r3.

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

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

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

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

The MRs MR4, MR3 and MR1 are associated with the route r4. *Delay_ r4*, *un_BW_r*4, *PL_r*4 and *Cost_r*4 are the 4 parameter values corresponding to the route r4.

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

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

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

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

2.3.2 Route Selection Algorithm:

The algorithm for the selection of the best route for data, voice and video related service is considered for discussion in this section.

Route selection for data service:

The proposed algorithm selects the route having maximum unused bandwidth for data service. LFN computes a parameter value r_data as r_data= (*un_BW_r*1 ˇ *un_BW_r*2 ˇ *un_BW_r*3 ˇ *un_BW_r*4). If r_data=*un_BW_r*1, r1 is the best route for data service. Similarly if r_data=*un_BW_r*2 or r_data = *un_BW_r*3 or r_data=*un_BW_r*4, r2 or r3 or r4 respectively is the best route for data service.

Route selection for voice service:

The proposed algorithm selects the route having minimum delay for voice service. LFN computes a parameter value r_voice as r_voice= (*Delay_r*1 ˆ *Delay_r*2 ˆ *Delay_r*3 ˆ *Delay_r*4). If r_voice=*Delay_r*1, r1 is the best route for voice service. Similarly if r_voice=*Delay_r*2 or r_voice=*Delay_r*3 or r_voice=*Delay_r*4, r2 or r3 or r4 respectively is the best route for voice service.

Route selection for video service:

The proposed algorithm selects the route having minimum packet loss for video service. LFN computes a parameter value r_video as r_video= (*PL_r*1 ˆ *PL_r*2 ˆ *PL_r*3 ˆ *PL_r*4). If r_video=*PL_r*1, r1 is the best route for video service. Similarly if r_video=*PL_r*2 or r_video=*PL_r*3 or r_video=*PL_r*4, r2 or r3 or r4 respectively is the best route for video service.

III. PROPOSED SECURITY ISSUE AND ITS SOLUTION

The security issue of the proposed route selection algorithm and its solution is discussed in this section.

3.1 Flooding of the LFN Queue:

The MNN sends MNN_LFN message to LFN for its session initiation. An attacker MNN may send a lot of MNN_LFN message to LFN intentionally to flood the LFN queue. But LFN is a fixed local infrastructure inside the NEMO. It is a very smart device, capable of handling large number of requests and possessing a large computational power. So it would be extremely difficult for a single MNN to inactive the LFN by sending requests. Thus in reality the mechanism of Distributed Denial of Service (DDoS) attack is considered. The main culprit MNN makes a number of MNNs to obey its order by taking them under its control. Then it manipulates all the MNNs to act as traps for other MNNs and so on. The MNNs which are converted to slave machines are termed as Zombies[9]. After a certain time a huge number of MNNs become Zombies. All these MNNs are used to send requests to the LFN which in turn fills up the LFN queue and prevents the other authentic MNNs from accessing the service of the LFN. This results in the obvious consequence of the LFN service failure which in turn increases session loss and reduces throughput of the network. Thus the simulation experiment is conducted to observe the variation of throughput and session loss vs. simulation time in the presence of attacker MNN as discussed in section 5.1.

3.2 Proposed Solution of Security Issue:

In this section the solution of the security issue as discussed in section 3.1 is considered for discussion.

3.2.1 Priority queue based solution:

In this method the LFN assigns a priority value to each request of MNN. If the number of requests from a particular MNN increase the priority value assign to each such service 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 LFN queue by the requests from malicious MNN. But this method is useful only if the LFN can recognize the MNN_id distinctly. But in case of DDoS attack the requests come from different MNNs making it practically impossible to stop this attack by this method.

3.2.2 Round Robin Method:

A non preemptive Round Robin Scheduling Algorithm is used to schedule the requests at the LFN queue. This process of implementation provides an equal amount of time to each MNN by using time-slicing technique.

IV. NEMO_SIM SIMULATOR

The proposed work is simulated with the help of a NEMO_SIM simulator. It is an application based object oriented simulator. This simulator is a software which takes a NEMO as input and produces performance measurement of NEMO as output. 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 program 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 [10], which is a JAVA Virtual Machine for NS2. NEMO_SIM can also act as an extended part of JNS 1.7, JAVA Network Simulator [11]. The NEMO in the proposed scheme is the combination of some interconnected processing units such as MNN, LFN, MR. Each processing unit is treated as thread 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.

4.1 MNN_REQ Thread:

It sends MNN_LFN message to LFN. A MNN has only one such thread.

4.2 LFN_MNN Thread:

It receives MNN_LFN message request from MNN, runs the route selection algorithm and sends LFN_MNN message to MNN.

4.3 MNN_SERVICE_START Thread:

It receives LFN_MNN response message and starts a new session. A MNN has only one such thread.

4.4 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.

4.5 MR_ROUTE_UPDATION Thread:

It sends MR_LFN message.

4.6 LFN_MR Thread:

It receives MR_LFN message.

4.7 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.

4.8 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.

4.9 MR_EGRESS_PACKET_LOSS Thread:

It discards the packets from the egress queue due to time out.

4.10 MNN_service_stop thread:

It receives LFN_MNN message but does not start a new session. Instead it discards the LFN_MNN messages.

4.11 False_MNN_service thread:

It generates spurious packets using the previous LFN_MNN response message and starts transmitting it through the desired route.

V. 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.

5.1 Simulation results with LFN flooding attack:

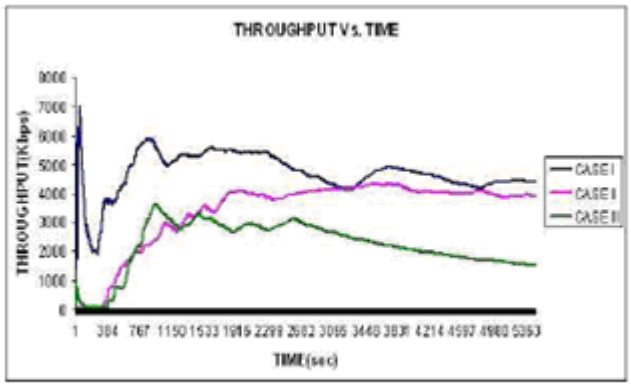

Fig.3 Throughput vs. Time for LFN Flooding Attack

The simulation experiment is conducted for 3 different cases as follows:

CASE I: No attacker MNN is present

CASE II: Only one MNN is an attacker MNN

CASE III: Three MNNs are attacker MNN

Fig.3 shows the plot of throughput vs. simulation time for all the 3 cases. The throughput is maximum in CASE I and minimum in CASE III. The throughput in CASE II is lesser than CASE I and higher than CASE III. Fig.4 shows the plot of session loss vs. simulation time for all the 3 cases. The session loss is minimum in CASE I and maximum in CASE III. The session loss in CASE II is lesser than CASE III and higher than CASE I.

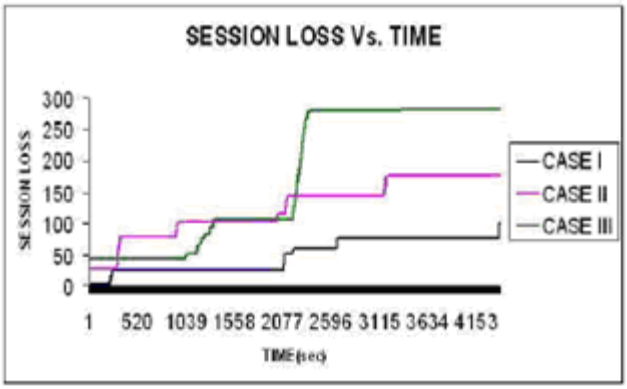

Fig.4 Session Loss vs. Time for LFN Flooding Attack

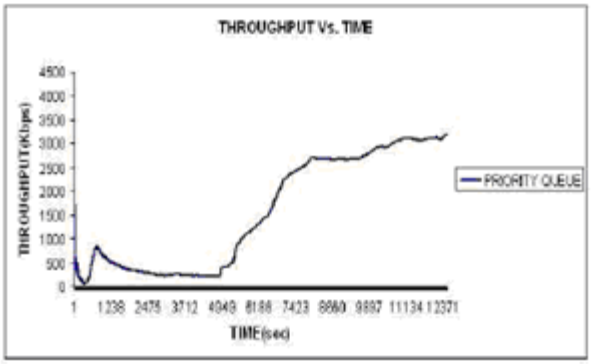

Fig.5 Throughput vs. Time with Priority Based Solution

Fig.6 Throughput vs. time with Round Robin Solution

5.2 Simulation result after incorporating solutions:

The simulation experiment is carried out considering the presence of 3 attackers MNN in MN. Fig.5 shows the plot of throughput vs. simulation time of the NEMO after incorporating priority queue based solution. It can be observed from Fig.5 that throughput increases with simulation time. Moreover it is higher than the throughput corresponding to CASE III in Fig.3. Fig.6 shows the plot of throughput vs. simulation time of the NEMO after incorporating round robin scheduling. It can be observed from Fig.6 that throughput increases

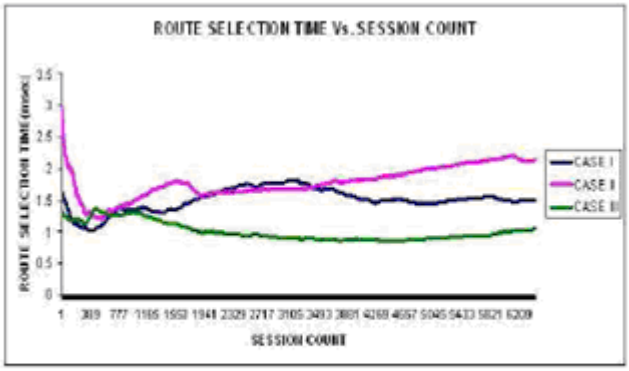

Fig.7 Route Selection Time vs. Session count with and without Solution

with simulation time. Moreover it is higher than the throughput corresponding to CASE III in Fig.3.

5.3 Dynamic behavior of the route selection time parameter after incorporating the solutions:

Fig.7 shows the variation of route selection time when the scheduling of the LFN queue is round robin type (CASEI), when the LFN queue is

implemented as priority queue (CASE II) and the LFN queue is implemented as a first in first out queue (CASE III). In CASE III the time complexity of placing a request in the queue and of removing a request from the queue is O (1). In CASE I the time complexity of placing a request in the queue is O(1) but the time complexity of removing a request from the queue on an average case is O(n) where n is the number of requests present in the LFN queue. In CASE II the priority queue is implemented as a heap. So the total time complexity of placing a request in the queue and of removing a request from the queue is O ($nlog_2n$), which is greater than both O (1) and O (n). From the plot it can be observed that the route selection time is maximum in CASE II and minimum in CASE III. The route selection time in CASE I is lesser than CASE II and higher than CASE III.

VI. CONCLUSION

This paper has presented the security issue of a route selection algorithm in a multihomed mobile network. 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 HA 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-wakikawamip6-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-wakikawamobileip-multiplecoa-05, February 2006.
8. T.Ernst and J.Charbon, ’Multihoming with NEMO Basic Support’,ICMU, Yokosuka, Japan, January 2004.
9. S.Gibson, ’Distributed Reflection Denial of Service’, Gibson Research Corporation, http://grc.com/dos/drdos.htm.
10. 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.
11. Java Network Simulator: http://jns.sourceforge.net.