Wireless Netw
DOI 10.1007/s11276-010-0274-y

Fuzzy logic based route optimization in a multihomed mobile networks

Sulata Mitra · Sumanta Pyne

Abstract The present work is a fuzzy logic based route optimization in multihomed mobile network. The mobile routers in the mobile network use best egress determination algorithm to identify its best egress interface for each service type supported by the mobile network dynamically and send the best egress interface per service type information to a local fixed node inside the mobile network. The mobile network node sends a request message to the local fixed node inside the mobile network to initiate a session. The local fixed node uses best route selection algorithm to select an optimal route from mobile network to Internet for the desired service type of the mobile network node. The performance of the proposed work is evaluated using NEMO_SIM simulator which is implemented using JAVA. Results based on a detailed performance evaluation study are also presented to demonstrate the efficacy of the proposed scheme.

Keywords Multihomed mobile network · Fuzzy logic · Route optimization · Multimedia traffic

1 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 multihoming support would enhance the load sharing and fault tolerant capabilities of mobile networks.

The existing node mobility arrangement protocols, like MIP protocols [1, 2] cann’t 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

S. Mitra (✉) · S. Pyne
Department of Computer Science & Technology, Bengal Engineering and Science University, Howrah, India
e-mail: mitra_sulata@hotmail.com

Published online: 08 September 2010

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.

In [8], the authors proposed a policy based routing protocol. It extends the prefixes scope binding update (PSBU) message to carry sufficient topology information about nested mobile network to HA. The binding associates the network prefix with the mobile router's CoA and a sequence of intermediated mobile router's CoAs. The mobile network prefix identifies the home link within the Internet topology. The same IP prefix is used by all the mobile network nodes. The CoAs of the MRs are the address of the intermediated hops during packet routing into the mobile network. The MR will send a PSBU message with a chain of CoAs to register with HA and core network (CN). The HA and CN builds a binding entry in their binding cache after receiving this message. The HA and CN send packets to mobile networks using an optimal routing path. The proposed routing protocol helps to achieve high throughput.

The present work (Fig. 1) considers (n,1,1) [9] configuration of MN. The proposed MN has 6 number of mobile routers (MRs) (NO_OF_MR = 6), single home agent (HA) and single mobile network prefix (MNP). The number of egress interface (NO_OF_EI) of each MR is assumed as 4 (NO_OF_EI = 4). A local fixed node (LFN) inside MN uses dynamic route selection algorithm to select an optimal route for 3 different service types (data, voice, and video) of mobile network node (MNN) independently. There are 4 possible number of routes (NO_OF_R = 4) of transmission from MN to Internet as shown in Table 1.

The scheme uses fuzzy logic based route selection algorithm to select an optimal route from MN to Internet.

The selection of route r1 depends upon the combined egress interface status of MR2 and MR1, selection of route r2 depends upon the combined egress interface status of MR4 and MR3, selection of route r3 depends upon the combined egress interface status of MR6 and MR5, selection of route r4 depends upon the combined egress interface status of MR4, MR3 and MR1. A new simulator (NEMO_SIM) is proposed in the present work and is used to study the performance of the proposed MN (Fig. 1).

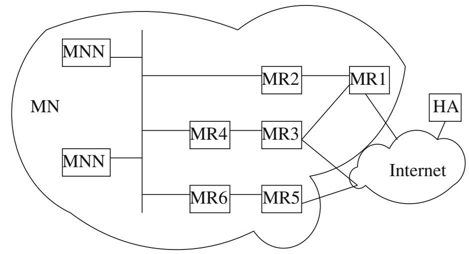

Fig. 1 Mobile network

Table 1

Route	Path	Route length in hops
r1	MNN → MR2 → MR1 → Internet	3
r2	MNN → MR4 → MR3 → Internet	3
r3	MNN → MR6 → MR5 → Internet	3
r4	MNN → MR4 → MR3 → MR1 → Internet	4

The main objective of the proposed scheme is to select the optimal route for the MNN from MN to Internet for their desired service type. The MRs in the MN always inform the LFN about the current status of their egress interface for each service type independently which helps LFN to select the optimal route for each service type of each MNN dynamically and independently. So at any instant of time the best possible route is selected for the desired service type of MNN. Such dynamic route selection algorithm must be capable of making a decision based on incomplete information and in a region of uncertainty. Fuzzy logic can be viewed as a theory for dealing with uncertainty about complex systems and as an approximation theory. In the present work the route selection algorithm exploits fuzzy logic to select optimal route for each service type supported by the MN independently.

2 Present work

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 and $1 \leq X \leq$ NO_OF_EI, the set Y indicates 4 parameters (NO_OF_P = 4) to determine the status of each egress interface and $1 \leq Y \leq$ NO_OF_P, the set Z indicates 3 different service types (NO_OF_ST = 3) that are supported by the MN and $1 \le Z \le NO\_OF\_ST$. Each MR maintains the values of the parameters such as Delay, Unused bandwidth, Packet loss and Cost corresponding to the 4 egress interfaces in the form of g(X, Y). It is defined as

$$\begin{bmatrix} g(X1,Y1) & g(X1,Y2) & g(X1,Y3) & g(X1,Y4) \\ g(X2,Y1) & g(X2,Y2) & g(X2,Y3) & g(X2,Y4) \\ g(X3,Y1) & g(X3,Y2) & g(X3,Y3) & g(X3,Y4) \\ g(X4,Y1) & g(X4,Y2) & g(X4,Y3) & g(X4,Y4) \end{bmatrix} = \begin{bmatrix} Delay_{E1} & un\_BW_{E1} & PL_{E1} & Cost_{E1} \\ Delay_{E2} & un\_BW_{E2} & PL_{E2} & Cost_{E2} \\ Delay_{E3} & un\_BW_{E3} & PL_{E3} & Cost_{E3} \\ Delay_{E4} & un\_BW_{E4} & PL_{E4} & Cost_{E4} \end{bmatrix}$$

where $Delay_{Ej}$, un_BW_{Ej}, PL_{Ej}, $Cost_{Ej}$ are the Delay, Unused bandwidth, Packet loss and Cost respectively of *j*th egress interface (Ej) of a MR where $1 \le j \le NO\_OF\_EI$.

The parameter value Y1 is in msec, Y2 is in kbps, Y3 is in % and Y4 is in unit. The minimum allowable delay among the allowable delays of all data, voice and video related services [10] is 250, 150 and 150 ms, respectively. The maximum required bandwidth to maintain data, voice and video related services [10] is 28.8, 21 and 32 kbps, respectively. All the services must be lossless in the best case. So the minimum packet loss for data, voice and video related services are assumed as 0, 1 and 1%. The minimum cost of each service type is assumed as the product of average packet size of any service type and cost/bit. The size of data, voice and video type of packet is assumed as 8000, 640 and 720 bits, respectively. The cost/bit is assumed as 1 unit. So the minimum cost for data, voice and video service types are 8000, 640 and 720 unit, respectively. In the best possible egress interface selection function and best possible route selection function for data related service the availability of bandwidth is assumed as most important to achieve fast data transfer and the packet loss should be minimum to achieve lossless data transfer. The data related service is assumed as delay insensitive. The voice related service is assumed as delay sensitive. So delay and packet loss should be minimum. It needs moderate bandwidth. The video related service is assumed as lossless and delay sensitive. It needs moderate bandwidth.

Each MR maintains the minimum allowable values of Delay, Packet loss, Cost and maximum allowable values of Desired bandwidth corresponding to the three different service types in the form of h(Y, Z). It is defined as

$$\begin{bmatrix} h(Y1,Z1) & h(Y1,Z2) & h(Y1,Z3) \\ h(Y2,Z1) & h(Y2,Z2) & h(Y2,Z3) \\ h(Y3,Z1) & h(Y3,Z2) & h(Y3,Z3) \\ h(Y4,Z1) & h(Y4,Z2) & h(Y4,Z3) \end{bmatrix} = \begin{bmatrix} 250 & 150 & 150 \\ 28.8 & 21 & 32 \\ 0 & 1 & 1 \\ 8000 & 640 & 720 \end{bmatrix}$$

Two fuzzifiers F1 and F2 are also maintained by each MR. F1 transforms g(X, Y) into the binary fuzzy relation P(X, Y) on fuzzy sets X and Y if at least one element of g(X, Y) changes. P(X, Y) is defined as

$$\begin{bmatrix} F1\_De_{E1} & F1\_BW_{E1} & F1\_PL_{E1} & F1\_cost_{E1} \\ F1\_De_{E2} & F1\_BW_{E2} & F1\_PL_{E2} & F1\_cost_{E2} \\ F1\_De_{E3} & F1\_BW_{E3} & F1\_PL_{E3} & F1\_cost_{E3} \\ F1\_De_{E4} & F1\_BW_{E4} & F1\_PL_{E4} & F1\_cost_{E4} \end{bmatrix}$$

where each element in P(X, Y) is the fuzzy value of the corresponding element in g(X, Y).

F2 transforms h(Y, Z) into the binary fuzzy relation Q(Y, Z) on fuzzy sets Y and Z only once when the system starts functioning as the value of all the elements in h(Y,Z) is constant. Q(Y,Z) is defined as

$$\begin{bmatrix} F2\_De_{Z1} & F2\_De_{Z2} & F2\_De_{Z3} \\ F2\_BW_{Z1} & F2\_BW_{Z2} & F2\_BW_{Z3} \\ F2\_PL_{Z1} & F2\_PL_{Z2} & F2\_PL_{Z3} \\ F2\_cost_{Z1} & F2\_cost_{Z2} & F2\_cost_{Z3} \end{bmatrix}$$

where each element of Q(Y, Z) is the fuzzy value of the corresponding element in h(Y, Z).

Each MR computes the fuzzy relation R(X, Z) using the max-min composition of the fuzzy relations P(X, Y) and Q(Y, Z). The min operation is used to determine the status of the *j*th egress interface per parameter for *Z*th service type in the worst case and the max operation is used to determine the status of the *j*th egress interface in the best case. P(X, Y) contains the fuzzy values corresponding to the 4 parameter values per egress interface and Q(Y, Z) contains the fuzzy values corresponding to the 4 parameter values per service type. R(X, Z) contains the fuzzy values to indicate the status of each egress interface per service type. So R(X, Z) = P(X, Y). Q(Y, Z). R(X, Z) is computed if at least one element of P(X, Y) changes. Each MR also determines the best possible egress interface for each service type independently if at least one element of R(X, Z) changes using best egress determination algorithm and sends this information to LFN.

LFN maintains 4 data structure and generates NO_OF_R number of routing table when the system starts functioning as discussed in Sect. 2.3.1 and 2.3.2 respectively. It updates the data structure and routing table after receiving best egress interface per service type information from MR. LFN also uses best route selection algorithm to select an optimal route after receiving a request message to initiate a session from MNN. Each session has a unique session identification (Session_id) number as assigned by LFN after selecting the optimal route. LFN maintains one counter (session_count) to count the number of active session. The counter value increases by 1 after selecting each optimal route per session.

2.1 Message exchange among various nodes of MN

The message exchange among various nodes of MN is shown in Fig. 2. When a MNN wants to initiate a session, it sends MNN_LFN message to LFN for the selection of an optimal route as source MNN. LFN uses best route selection algorithm to determine the best possible route from MN to Internet for the desired service type of the MNN. LFN returns the optimal route along with the Session_id to the source MNN in the form of LFN_MNN message.

The source MNN initiates the session and generates MNN_MR message after receiving LFN_MNN message. The MNN_MR message is used as the header of the packets corresponding to the desired service type of source MNN as discussed in Sect. 2.1.3. Each MR generates MR_LFN message and sends it to LFN after executing best egress determination algorithm.

2.1.1 MNN_LFN message

It contains 3 component as MNN identification (MNN_id), service type (S_type), request number. 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. If the total number of request per MNN is req_no, the number of bits require to represent request number is $\log_2$(req_no). So the length of MNN_LFN message is $19 + \log_2$(req_no) bits.

2.1.2 LFN_MNN message

This message contains Session_id, l_i-1 number of (MR_id,E_id) pairs where l_i is the length of *i*th route in number of hops and $1 \leq i \leq$ NO_OF_R. The MR_id (MR identification) component of each pair uses best egress determination algorithm to determine the best egress interface for the desired service type of MNN. The E_id (egress identification) component of the same pair is the identification number of the best egress interface as determined by MR_id component.

For example let the route r1 is selected as optimal route for the desired service type of MNN by the best route selection algorithm. MR1 and MR2 are the MR_id of the MRs which are associated with the route r1. So this message contains two pair as (MR2, E_id of suitable egress interface of MR2), (MR1, E_id of suitable egress interface of MR1). The E_id of suitable egress interface of MR1 and E_id of suitable egress interface of MR2 are determined by MR1 and MR2 using best egress determination algorithm. LFN generates Session_id and l_i-1 number of (MR_id, E_id) pairs of this message with computation complexity O(l_i). In case of 6 MRs, the number of bits requires to represent MR identification (MR_id) is 3. In case of 4 egress interfaces of each MR, the number of bits require to represent egress identification (E_id) is 2. So the number of bits require to represent a single (MR_id, E_id) pair is 5. In case of the routes r1, r2, r3 LFN_MNN message contains 2 such pair as 2 MR is associated with these routes. So the length of LFN_MNN message is the sum of $\log_2$(session_count) and 10 bits for 2 (MR_id, E_id) pair. In case of route r4 LFN_MNN message contains 3 such pair as 3 MR is associated with r4. So the length of LFN_MNN message is the sum of $\log_2$(session_count) and 15 bits for 3 (MR_id, E_id) pair.

2.1.3 MNN_MR message

The MNN_MR message has 3 different formats (Fig. 3). The format as shown in Fig. 3a is used as the header of the first packet, the format as shown in Fig. 3b is used as the header of the last packet and the format as shown in Fig. 3c is used as the header for all the intermediate packets corresponding to the desired service type of MNN.

MNN_id indicates source identification of the session.

S_no indicates the sequence number of each packet.

P_no indicates the number of packet in the corresponding session.

S_flag indicates start flag. It is set in the first packet of the session to indicate the start of the session.

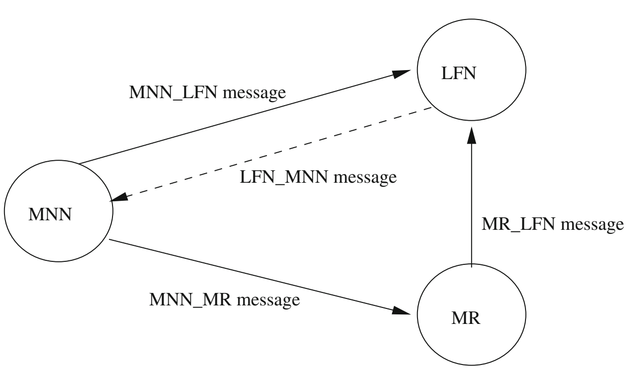

Fig. 2 Message exchange among various nodes of mobile network

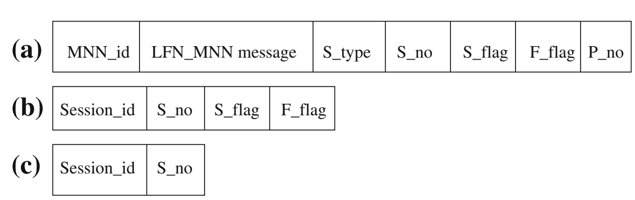

Fig. 3 MNN_MR message format **a** Packet header for len_first, **b** Packet header for len_last, **c** Packet header for len_int

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

The length of the header in the first packet (len_first) is ($\log_2$(MNN_id*S_type*S_no*S_flag*F_flag*P_no) + length of LFN_MNN message) bits. The length of the header in the last packet (len_last) is $\log_2$(session_count*S_no*S_flag*F_flag) bits. The length of the header in all the intermediate packets (len_int) is $\log_2$(session_count*S_no) bits.

2.1.4 MR_LFN message

Each MR generates this message after executing best egress determination algorithm. The format of this message which is generated by *p*th MR is shown in Fig. 4. This message has 3 parts. The part1 of this message (Fig. 4a) contains 3 services Z1, Z2 and Z3 along with their identification number 0, 1 and 2, respectively. The identification number for service type Z1 is 0. So the *0*th element of the variable array (variable_array [0]) in part3 of the message contains the 4 parameter values (i.e. NO_OF_P) (Y1, Y2, Y3, Y4) of the best egress interface which is determined by *p*th MR using best egress determination algorithm for service type Z1 in the form gp(X, Y). Similarly the *1*th element (variable_array [1]) and the *2*th element (variable_array [2]) of the variable array in part3 of the message contain 4 parameter values of the best egress interface which is determined by *p*th MR using best egress determination algorithm for service type Z2 and Z3 respectively. For 3 services the number of bits requires to represent each identification number is 2. So the length of part1 is 6 bits.

The part2 of this message (Fig. 4b) contains fuzzy values corresponding to the best egress interface of *p*th MR per service type. The egress interfaces X1, X2 and X3 of *p*th MR are assumed as the best egress interfaces for service type Z1, Z2 and Z3, respectively. So the part2 of the message format in Fig. 4b indicates X1 is the best egress interface of *p*th MR for service type Z1, X2 is the best egress interface of *p*th MR for service type Z2 and X3 is the best egress interface of *p*th MR for service type Z3. MRp(X1, Z1), MRp(X2, Z2) and MRp(X3, Z3) are the 3 fuzzy values as determined by *p*th MR during the execution of the best egress determination algorithm as discussed in Sect. 2.2.5. The size of each fuzzy value is assumed as 32 bits. So the length of part2 is $3 \times 32 = 96$ bits.

The part3 of the MR_LFN message (Fig. 4c) is an array having variable number of elements and so this array is known as variable array. Each element of this array indicates the best egress interface in the form X and its 4 parameter values (i.e. NO_OF_P) in the form gp(X, Y) (it is g(X, Y) as computed by *p*th MR) for *p*th MR per service type. So each element contains (NO_OF_P + 1) number of components. The number of elements in this array is decided depending upon 3 different cases. In the worst case the best egress interface of 3 different service types as determined by *p*th MR using the best egress determination algorithm are different i.e. $X1 \neq X2 \neq X3$. In the average case the best egress interface of any 2 different service types as determined by *p*th MR using the best egress determination algorithm are identical i.e. ($X1 = X2$ and $X1 \neq X3$ and $X2 \neq X3$) or ($X1 = X3$ and $X1 \neq X2$ and $X2 \neq X3$) or ($X2 = X3$ and $X1 \neq X3$ and $X1 \neq X2$). In the best case the best egress interface of 3 different service types as determined by *p*th MR using the best egress determination algorithm are identical i.e. ($X1 = X2 = X3$). So the number of elements in the variable array for the worst case is 3 (i.e. NO_OF_ST) one per service type, for the average case is 2 and for the best case is 1. For example, the *0*th element of variable_array (variable_array [0]) in Fig. 4c contains X1 as the best egress identification number of *p*th MR for service type Z1. It also contains gp(X1, Y1), gp(X1, Y2), gp(X1, Y3) and gp(X1, Y4) as the 4 parameter values of X1 egress interface. For 4 egress interfaces per MR the number of bits require to represent each egress identification number is 2. The size of each gp(X, Y) is assumed as 32 bits. So the length of part3 in the worst case is $3*(2 + 4*32) = 390$ bits. The length of the MR_LFN message is $6 + 3*32 + 390 = 492$ bits. The length of part3 in the average case is $2*(2 + 4*32) = 260$ bits. The length of the MR_LFN message is 362 bits. The length of part3 in the best case is $1*(2 + 4*32) = 130$ bits. So the length of the MR_LFN message is 232 bits. The computation complexity to generate MR_LFN message depends on the computation complexity of generating

Z1(0), Z2(1), Z3(2)

(a)

MRp(X1,Z1), MRp(X2,Z2), MRp(X3,Z3)

(b)

variable_array[0]

X1,gp(X1,Y1),gp(X1,y2),gp(X1,Y3),gp(X1,Y4)

variable_array[1]

X2,gp(X2,Y1),gp(X2,y2),gp(X2,Y3),gp(X2,Y4)

variable_array[2]

X3,gp(X3,Y1),gp(X3,y2),gp(X3,Y3),gp(X3,Y4)

(c)

Fig. 4 **a** Part1 of MR_LFN message. **b** Part2 of MR_LFN message. **c** Part3 of MR_LFN message part3 of this message due to its variable length. MR generates a single element having (NO_OF_P + 1) number of components in the best case with computation complexity O(NO_OF_P) and (NO_OF_ST) number of elements each having (NO_OF_P + 1) number of components in the worst case with computation complexity O(NO_OF_ST*NO_OF_P).

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 (Table 2) to keep the records of various sessions in the form (MNN_id, E_id, Session_id, desire_BW, Next_hop). One record is maintained for each session of a MNN. Each MR associated with the optimal route which is selected for the desired service type of MNN by the best route selection algorithm 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. The value of the attributes MNN_id and Session_id of each record are obtained from the header available with the first packet of the corresponding session (Fig. 3a). The desired bandwidth (desire_BW) attribute of each record indicates the required bandwidth to maintain the desired service type. It is assumed that each MR knows the desired bandwidth for each of the 3 service type that is supported by the MN. The next hop (Next_hop) attribute of each record indicates the next node associated with the optimal route corresponding to the desired service type. The next hop attribute is Internet in the routing table which is maintained by the root MR associated with the optimal route. It is MR in the routing table which is maintained by the leaf MR and by all the intermediate MRs associated with the optimal route. The LFN_MNN message part of the header available with the first packet (Fig. 3a) of the corresponding session is used by all the MRs associated with the optimal route to insert the value of the attributes E_id and Next_hop in a record. For example, if the optimal route is r1, MR2 is the leaf MR and MR1 is the root MR (Fig. 1). The LFN_MNN message part in the header of the first packet contains (MR2,E_id of suitable egress interface of MR2), (MR1, E_id of suitable egress interface of MR1). MR2 inserts "E_id of suitable egress interface of MR2" as the E_id attribute and MR1 as the Next_hop attribute in the corresponding record. MR1 inserts "E_id of suitable egress interface of MR1" as the E_id attribute and Internet as the Next_hop attribute in the corresponding record. When a MR receives packet from a MNN, it searches the routing table using Session_id as the searching key to retrieve the record of MNN with computation complexity O(1). The value of Session_id is obtained from the header of each packet. The MR retrieves the value of the attribute E_id from the record of MNN to deliver the packet to the desired egress interface and the value of the attribute Next_hop from the record of MNN to deliver the packet to the next hop of the optimal route.

The record for *i*th MNN (MNN_i), *j*th egress interface (Ej), *S*th session (S_s) is shown in Table 2. BW_Z indicates the desired bandwidth for *Z*th service type.

Table 2

MNN_id	E_id	Session_id	desire_BW	Next_hop
MNN_i	Ej	S_s	BW_Z	MR/Internet

2.2.2 Operation of F1

F1 transforms g(X, Y) to P(X, Y).

Computation of g(X, Y): The input parameters of F1 are the 4 parameter values (Y1, Y2, Y3, Y4) of 4 egress interfaces (X1, X2, X3, X4) in the form g(X, Y). The computation of 4 parameter values for jth egress interface where $1 \leq j \leq NO\_OF\_EI$ is discussed below.

When a MNN sends a packet to the ingress interface of the leaf MR associated with the optimal route as selected by the best 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 through the selected egress interface. The difference of the two time stamp (δt) is considered as the delay per packet for that MNN. The initial value of delay at Ej ($Delay_{Ej}$) is assumed as 0.0 ms. 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 with computation complexity O(1). The *p*th MR computes gp(X1, Y1), gp(X2, Y1), gp(X3, Y1) and gp(X4, Y1) for egress interfaces X1, X2, X3 and X4, respectively.

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 of *i*th MNN, un_BW_{Ej} is increased by $desire\_BW_{ij}$ with computation complexity O(1). The value of $desire\_BW_{ij}$ is obtained from the desire_BW attribute of the record corresponding to MNN_i from the routing table (Table 2). The *p*th MR computes gp(X1, Y2), gp(X2, Y2), gp(X3, Y2) and gp(X4, Y2) for egress interfaces X1, X2, X3 and X4, respectively.

The packet loss at any egress interface is the sum 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 with computation complexity O(number of packets in egress buffer). PL_{Ej} is also increased by 1 after removing a packet from the buffer at Ej due to buffer overflow with computation complexity O(1). The packet loss at Ej is computed in % as (PL_{Ej}/total packet at Ej)*100. The *p*th MR computes gp(X1, Y3), gp(X2, Y3), gp(X3, Y3) and gp(X4, Y3) for egress interfaces X1, X2, X3 and X4, respectively.

The cost per egress interface is the sum of cost of all the MNN using that particular egress interface. The cost of each MNN is the sum 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 sum 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 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 $Cost_{ij} = [(34 + \log_2(req\_no) + \log_2(session\_count) + len\_first + len\_last + ((P\_no-2) * len\_int) + (P\_no * P\_sz)] * cost/bit$, where the value of P_{no} is obtained from the packet header of the first packet corresponding to the desired service type. So after receiving the first packet from *i*th MNN, $Cost_{Ej}$ is increased by $Cost_{ij}$ with computation complexity O(1). The *p*th MR computes gp(X1, Y4), gp(X2, Y4), gp(X3, Y4) and gp(X4, Y4) for egress interfaces X1, X2, X3 and X4, respectively. Each MR performs the same computation to calculate the 4 parameter values (Delay, Unused bandwidth, Packet loss, Cost) of all its 4 egress interfaces. Each MR also maintains a table (Table 3) to keep the parameter values of 4 egress interfaces in the form (Egress, Delay, Unused bandwidth, Packet loss, Cost). The parameter values of *j*th egress interface are shown in Table 3.

Table 3

Egress	Delay	Unused bandwidth	Packet loss	Cost
Ej	$Delay_{Ej}$	un_BW_{Ej}	PL_{Ej}	$Cost_{Ej}$

Computation of P(X, Y): The output parameters of F1 are the 4 fuzzy values of 4 parameters per egress interface in the form P(X,Y). The computation of the 4 fuzzy values at the output of F_1 for Ej where $1 \le j \le NO\_OF\_EI$ using the appropriate fuzzy membership function is discussed below.

$$F1\_De_{Ej} = 1 - \left(\mathrm{Delay}_{Ej} \Big/ \sum_{j=1}^{NO\_OF\_EI} \mathrm{Delay}_{Ej}\right) \; if \; \left(\sum_{j=1}^{NO\_OF\_EI} \mathrm{Delay}_{Ej}\right) \neq 0$$

$$F1\_De_{Ej} \text{ is assumed as 1 if } \left(\sum_{j=1}^{NO\_OF\_EI} \mathrm{Delay}_{Ej}\right) = 0$$

$\mathrm{Delay}_{Ej} \Big/ \sum_{j=1}^{NO\_OF\_EI} \mathrm{Delay}_{Ej}$ indicates delay at Ej w.r.t. NO_OF_EI number of egress interfaces. If $Delay_{Ej}$ is very high, $F1\_De_{Ej}$ becomes very low which indicates the status of *j*th egress interface is not good for the parameter delay.

$F1\_De_{Ej}$ is computed using (NO_OF_EI-1) number of addition, 1 division and 1 subtraction with computation complexity O(NO_OF_EI). The F1 fuzzifier at *p*th MR computes $F1\_De_{E1}$, $F1\_De_{E2}$, $F1\_De_{E3}$ and $F1\_De_{E4}$ of its 4 egress interfaces E1, E2, E3 and E4, respectively.

$$F1\_BW_{Ej} = (un\_BW_{Ej})/(\text{availablebandwidth at Ej})$$

If un_BW_{Ej} is very high $F1\_BW_{Ej}$ is very close to 1 which indicates the status of *j*th egress interface is good for the parameter bandwidth.

$F1\_BW_{Ej}$ is computed with computation complexity O(1). The F1 fuzzifier at *p*th MR computes $F1\_BW_{E1}$, $F1\_BW_{E2}$, $F1\_BW_{E3}$ and $F1\_BW_{E4}$ of its 4 egress interfaces E1, E2, E3 and E4, respectively.

$$F1\_PL_{Ej} = 1 - (PL_{Ej})/(\text{totalpacket at Ej})$$

$$F1\_PL_{Ej} \text{ is assumed as 1 if (total packet at Ej) is zero.}$$

(PL_{Ej})/(total packet at Ej) indicates the probability of packet loss at Ej. If PL_{Ej} is very high, $F1\_PL_{Ej}$ is very low which indicates the status of *j*th egress interface is not good for the parameter packet loss.

$F1\_PL_{Ej}$ is computed with computation complexity O(1). The F1 fuzzifier at *p*th MR computes $F1\_PL_{E1}$, $F1\_PL_{E2}$, $F1\_PL_{E3}$ and $F1\_PL_{E4}$ of its 4 egress interfaces E1, E2, E3 and E4, respectively.

$$F1\_\cos t_{Ej} = 1 - \left(Cost_{Ej} \Big/ \sum_{j=1}^{NO\_OF\_EI} Cost_{Ej}\right) \text{ if } \left(\sum_{j=1}^{NO\_OF\_EI} Cost_{Ej}\right) \neq 0$$

$$F1\_\cos t_{Ej} \text{ is assumed as 1 if } \left(\sum_{j=1}^{NO\_OF\_EI} Cost_{Ej}\right) = 0$$

$Cost_{Ej} \Big/ \sum_{j=1}^{NO\_OF\_EI} Cost_{Ej}$ indicates cost at Ej w.r.t. NO_OF_EI number of egress interfaces. If $Cost_{Ej}$ is very high, $F1\_cost_{Ej}$ becomes very low which indicates the status of *j*th egress interface is not good for the parameter cost.

$F1\_cost_{Ej}$ is computed using (NO_OF_EI-1) number of addition, 1 division and 1 subtraction with computation complexity O(NO_OF_EI). The F1 fuzzifier at *p*th MR computes $F1\_cost_{E1}$, $F1\_cost_{E2}$, $F1\_cost_{E3}$ and $F1\_cost_{E4}$ of its 4 egress interfaces E1, E2, E3 and E4, respectively.

2.2.3 Operation of F2

F2 transforms h(Y, Z) to Q(Y, Z).

Computation of Q(Y, Z): The output parameters of F2 are the 4 fuzzy values of 4 parameters per service type in the form Q(Y, Z). The fuzzy values at the output of F2 indicates minimum allowable values of Delay, Packet loss, Cost and maximum allowable desired bandwidth corresponding to the 3 different service types (i.e. NO_OF_ST). The computation of 4 fuzzy values at the output of F2 for *Z*th service type, where $1 \leq Z \leq$ NO_OF_ST using the appropriate fuzzy membership function are discussed below.

$$F2\_De_Z = a'k/h(Y1, Z)$$

where a′k is computed as h(Y1, Z1)∧h(Y1, Z2)∧h(Y1, Z3) and is equal to 150 ms. So a′k is lesser than h(Y1, Z) and $F2\_De_Z$ lies within 0 to 1. The F2 fuzzifier at *p*th MR computes $F2\_De_{Z1}$, $F2\_De_{Z2}$, $F2\_De_{Z3}$ for service types Z1, Z2, Z3, respectively.

$$F2\_BW_z = h(Y2, z)/b'k$$

where b′k is computed as h(Y2, Z1)∨h(Y2, Z2)∨h(Y2, Z3) and is equal to 32 Kbps. So b′k is greater than h(Y2, Z) and $F2\_BW_Z$ lies within 0 to 1. The F2 fuzzifier at *p*th MR computes $F2\_BW_{Z1}$, $F2\_BW_{Z2}$, $F2\_BW_{Z3}$ for service types Z1, Z2, Z3, respectively.

$$F2\_PL_Z = 1 - h(Y3, Z)$$

h(Y3, Z) indicates minimum allowable packet loss for *Z*th service type in percentage whereas $F2\_PL_Z$ indicates how lossless the service type Z is. The F2 fuzzifier at *p*th MR computes $F2\_PL_{Z1}$, $F2\_PL_{Z2}$, $F2\_PL_{Z3}$ for service types Z1, Z2, Z3, respectively.

$$F2\_cost_Z = d'k/h(Y4, Z)$$

where d′k is computed as h(Y4, Z1)∧h(Y4, Z2)∧h(Y4, Z3) and is equal to 640 unit. So d′k is lesser than h(Y4, Z) and $F2\_cost_Z$ lies within 0 to 1.

a′k, b′k and d′k are computed using (NO_OF_ST-1) number of min, max and min operation, respectively with computation complexity O(NO_OF_ST). The F2 fuzzifier at *p*th MR computes $F2\_cost_{Z1}$, $F2\_cost_{Z2}$, $F2\_cost_{Z3}$ for service types Z1, Z2, Z3, respectively. $F2\_De_Z$, $F2\_BW_Z$, $F2\_PL_Z$ and $F2\_cost_Z$ are computed with computation complexity O(1).

2.2.4 Computation of R(X, Z)

R(X, Z) is the max-min composition of P(X, Y) and Q(Y, Z) as discussed below.

$$\begin{bmatrix} F1\_De_{E1} & F1\_BW_{E1} & F1\_PL_{E1} & F1\_cost_{E1} \\ F1\_De_{E2} & F1\_BW_{E2} & F1\_PL_{E2} & F1\_cost_{E2} \\ F1\_De_{E3} & F1\_BW_{E3} & F1\_PL_{E3} & F1\_cost_{E3} \\ F1\_De_{E4} & F1\_BW_{E4} & F1\_PL_{E4} & F1\_cost_{E4} \end{bmatrix} \begin{bmatrix} F2\_De_{Z1} & F2\_De_{Z2} & F2\_De_{Z3} \\ F2\_BW_{Z1} & F2\_BW_{Z2} & F2\_BW_{Z3} \\ F2\_PL_{Z1} & F2\_PL_{Z2} & F2\_PL_{Z3} \\ F2\_cost_{Z1} & F2\_cost_{Z2} & F2\_cost_{Z3} \end{bmatrix}$$

and is equal to

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \\ P_{41} & P_{42} & P_{43} \end{bmatrix}$$

where

P_{11} =$(F1\_De_{E1} \wedge F2\_De_{Z1}) \vee (F1\_BW_{E1} \wedge F2\_BW_{Z1}) \vee (F1\_PL_{E1} \wedge F2\_PL_{Z1}) \vee (F1\_cost_{E1} \wedge F2\_cost_{Z1})$

P_{12} =$(F1\_De_{E1} \wedge F2\_De_{Z2}) \vee (F1\_BW_{E1} \wedge F2\_BW_{Z2}) \vee (F1\_PL_{E1} \wedge F2\_PL_{Z2}) \vee (F1\_cost_{E1} \wedge F2\_cost_{Z2})$

P_{13} =$(F1\_De_{E1} \wedge F2\_De_{Z3}) \vee (F1\_BW_{E1} \wedge F2\_BW_{Z3}) \vee (F1\_PL_{E1} \wedge F2\_PL_{Z3}) \vee (F1\_cost_{E1} \wedge F2\_cost_{Z3})$

P_{21} =$(F1\_De_{E2} \wedge F2\_De_{Z1}) \vee (F1\_BW_{E2} \wedge F2\_BW_{Z1})$ $(F1\_PL_{E2} \wedge F2\_PL_{Z1}) \vee (F1\_cost_{E2} \wedge F2\_cost_{Z1})$

P_{22} =$(F1\_De_{E2} \wedge F2\_De_{Z2}) \vee (F1\_BW_{E2} \wedge F2\_BW_{Z2})$ $(F1\_PL_{E2} \wedge F2\_PL_{Z2}) \vee (F1\_cost_{E2} \wedge F2\_cost_{Z2})$

P_{23} =$(F1\_De_{E2} \wedge F2\_De_{Z3}) \vee (F1\_BW_{E2} \wedge F2\_BW_{Z3})$ $(F1\_PL_{E2} \wedge F2\_PL_{Z3}) \vee (F1\_cost_{E2} \wedge F2\_cost_{Z3})$

P_{31} =$(F1\_De_{E3} \wedge F2\_De_{Z1}) \vee (F1\_BW_{E3} \wedge F2\_BW_{Z1}) \vee (F1\_PL_{E3} \wedge F2\_PL_{Z1}) \vee (F1\_cost_{E3} \wedge F2\_cost_{Z1})$

P_{32} =$(F1\_De_{E3} \wedge F2\_De_{Z2}) \vee (F1\_BW_{E3} \wedge F2\_BW_{Z2}) \vee (F1\_PL_{E3} \wedge F2\_PL_{Z2}) \vee (F1\_cost_{E3} \wedge F2\_cost_{Z2})$

P_{33} =$(F1\_De_{E3} \wedge F2\_De_{Z3}) \vee (F1\_BW_{E3} \wedge F2\_BW_{Z3}) \vee (F1\_PL_{E3} \wedge F2\_PL_{Z3}) \vee (F1\_cost_{E3} \wedge F2\_cost_{Z3})$

P_{41} =$(F1\_De_{E4} \wedge F2\_De_{Z1}) \vee (F1\_BW_{E4} \wedge F2\_BW_{Z1}) \vee (F1\_PL_{E4} \wedge F2\_PL_{Z1}) \vee (F1\_cost_{E4} \wedge F2\_cost_{Z1})$

P_{42} =$(F1\_De_{E4} \wedge F2\_De_{Z2}) \vee (F1\_BW_{E4} \wedge F2\_BW_{Z2}) \vee (F1\_PL_{E4} \wedge F2\_PL_{Z2}) \vee (F1\_cost_{E4} \wedge F2\_cost_{Z2})$

P_{43} $=(F1\_De_{E4} \wedge F2\_De_{Z3}) \vee (F1\_BW_{E4} \wedge F2\_BW_{Z3}) \vee (F1\_PL_{E4} \wedge F2\_PL_{Z3}) \vee (F1\_cost_{E4} \wedge F2\_cost_{Z3})$

R(X, Z) has NO_OF_ST*NO_OF_EI number of elements. The computation of each element in R(X,Z) needs (NO_OF_P) number of min operations and (NO_OF_P-1) number of max operations. So each element of R(X,Z) is computed with computation complexity O(2*NO_OF_P-1).

2.2.5 Best egress determination algorithm

Each MR in the MN uses this algorithm to determine the best possible egress interface per service type supported by the MN independently. This algorithm for *p*th MR (MRp) and *Z*th service type is considered for discussion in this section.

For *p*th MR and *Z*th service type MRp(X, Z) = Rp(X1, Z)∨Rp(X2, Z)∨Rp(X3, Z)∨Rp(X4, Z), where Rp(X1, Z), Rp(X2, Z), Rp(X3, Z) and Rp(X4, Z) are the R(X, Z) fuzzy relation for *p*th MR and *Z*th service type. If MRp(X, Z) = Rp(X3, Z), X3 is the best possible egress interface of *p*th MR for *Z*th service type. The computation of MRp(X, Z) needs (NO_OF_EI-1) number of max operations and so it is computed with computation complexity O(NO_OF_EI). In the best case 1 egress interface of *p*th MR is determined as the best egress interfaces for *Z*th service type by the algorithm with computation complexity O(NO_OF_EI). In the worst case multiple egress interfaces of *p*th MR are determined as the best egress interface for *Z*th service type. In such a case the algorithm uses best possible egress interface selection function to determine the optimal egress interface of *p*th MR for *Z*th service type. Let (NO_OF_EI) number of egress interfaces of *p*th MR is selected as best egress interfaces for *Z*th service type. The algorithm uses best possible egress interface selection function for (NO_OF_EI-1) number of times to select the optimal egress interface of *p*th MR from (NO_OF_EI) number of egress interfaces for *Z*th service type depending upon the 4 parameter values (i.e. NO_OF_P), Y1, Y2, Y3, Y4 with computation complexity O(NO_OF_EI*NO_OF_P).

2.2.5.1 Best possible egress interface selection function Let the *q*th and *n*th interface (Xq and Xn) of *p*th MR (MRp) are selected as the best possible egress interface (X) for *Z*th service type (X = BestEgress(MRp,Z)) where ($1 \leq q \leq$ NO_OF_EI, $1 \leq n \leq$ NO_OF_EI and $q \neq n$). gp(Xq,Y1), gp(Xq,Y2), gp(Xq,Y3), gp(Xq,Y4) are the 4 parameter values of Xq and gp(Xn,Y1), gp(Xn,Y2), gp(Xn,Y3), gp(Xn,Y4) are the 4 parameter values of Xn.

```
if (Z=Z1)
{
 if (gp(Xq,Y2)=gp(Xn,Y2))
  {
   if (gp(Xq,Y3)=gp(Xn,Y3))
    {
     if (gp(Xq,Y1)=gp(Xn,Y1))
      {
       if (gp(Xq,Y4)=gp(Xn,Y4))
         randomly select one of Xq, Xn as BestEgress(MRp,Z)
       else
        {
         if (gp(Xq,Y4)>gp(Xn,Y4))
           BestEgress(MRp,Z)<-Xn
         else
           BestEgress(MRp,Z)<-Xq
        }
      }
     else
      {
       if (gp(Xq,Y1)>gp(Xn,Y1))
         BestEgress(MRp,Z)<-Xn
       else
         BestEgress(MRp,Z)<-Xq
      }
    }
   else
      {
       if (gp(Xq,Y3)>gp(Xn,Y3))
         BestEgress(MRp,Z)<-Xn
       else
         BestEgress(MRp,Z)<-Xq
      }
  }
else
      {
       if (gp(Xq,Y2)<gp(Xn,Y2))
         BestEgress(MRp,Z)<-Xn
       else
         BestEgress(MRp,Z)<-Xq
      }
  X=BestEgress(MRp,Z);
}
else
 if (Z=Z2)
{
 if (gp(Xq,Y1)=gp(Xn,Y1))
  {
   if (gp(Xq,Y3)=gp(Xn,Y3))
    {
     if (gp(Xq,Y2)=gp(Xn,Y2))
      {
       if (gp(Xq,Y4)=gp(Xn,Y4))
         randomly select one of Xq, Xn as BestEgress(MRp,Z)
       else
        {
         if (gp(Xq,Y4)>gp(Xn,Y4))
           BestEgress(MRp,Z)<-Xn
         else
           BestEgress(MRp,Z)<-Xq
        }
      }
     else
      {
       if (gp(Xq,Y2)>gp(Xn,Y2))
         BestEgress(MRp,Z)<-Xq
```

```
      else
       BestEgress(MRp,Z)<-Xn
      }
    }
   else
      {
      if (gp(Xq,Y3)>gp(Xn,Y3))
       BestEgress(MRp,Z)<-Xn
      else
       BestEgress(MRp,Z)<-Xq
      }
  }
 else
      {
       if (gp(Xq,Y1)>gp(Xn,Y1))
       BestEgress(MRp,Z)<-Xn
       else
       BestEgress(MRp,Z)<-Xq
      }
 X=BestEgress(MRp,Z);
}
else
 if (Z=Z3)
{
 if (gp(Xq,Y3)=gp(Xn,Y3))
   {
    if (gp(Xq,Y1)=gp(Xn,Y1))
     {
     if (gp(Xq,Y2)=gp(Xn,Y2))
      {
      if (gp(Xq,Y4)=gp(Xn,Y4))
       randomly select one of Xq, Xn as BestEgress(MRp,Z)
      else
       {
       if (gp(Xq,Y4)>gp(Xn,Y4))
        BestEgress(MRp,Z)<-Xn
       else
        BestEgress(MRp,Z)<-Xq
       }
      }
     else
      {
      if (gp(Xq,Y2)>gp(Xn,Y2))
       BestEgress(MRp,Z)<-Xq
      else
       BestEgress(MRp,Z)<-Xn
      }
    }
   else
      {
      if (gp(Xq,Y1)>gp(Xn,Y1))
       BestEgress(MRp,Z)<-Xn
      else
       BestEgress(MRp,Z)<-Xq
      }
  }
 else
      {
       if (gp(Xq,Y3)>gp(Xn,Y3))
       BestEgress(MRp,Z)<-Xn
       else
       BestEgress(MRp,Z)<-Xq
      }
   X=BestEgress(MRp,Z);
}
```

2.3 Function of LFN

The function of LFN is considered for discussion in the following sections.

2.3.1 Maintenance of data structure

The data structure 1 is MR_EG_STA. Each element of MR_EG_STA[p][j][Z] is a fuzzy value to indicate the status of *j*th egress interface of *p*th MR for *Z*th service type. So it is a 3-D array of dimension NO_OF_MR * NO_OF_EI * NO_OF_ST. All the elements of this 3-D array are initialized by 1.0 when the system starts functioning. LFN updates the element values of the MR_EG_STA data structure after receiving MR_LFN message. LFN updates the value of the 3 array elements (i.e. NO_OF_ST) one for each service type, MR_EG_STA[p][X1][Z1], MR_EG_STA[p][X2][Z2] and MR_EG_STA[p][X3][Z3] corresponding to the egress interfaces X1, X2 and X3 of *p*th MR using MRp(X1,Z1), MRp(X2,Z2) and MRp(X3,Z3), respectively with computation complexity O(NO_OF_ST). The fuzzy values corresponding to MRp(X1,Z1), MRp(X2,Z2) and MRp(X3,Z3) are obtained from the part2 of MR_LFN message as discussed in Sect. 2.1.4.

The data structure 2 is MR_EGRESS_INFORM. Each element of MR_EGRESS_INFORM[p][j][y] indicates the value of *y*th parameter at *j*th egress interface of *p*th MR. So it is a 3-D array of dimension NO_OF_MR*NO_OF_EI*NO_OF_P. For example, the value of the elements MR_EGRESS_INFORM[p][j][Y1], MR_EGRESS_INFORM[p][j][Y2], MR_EGRESS_INFORM[p][j][Y3] and MR_EGRESS_INFORM[p][j][Y4] indicate Delay (Y1), Unused bandwidth (Y2), Packet loss (Y3) and Cost (Y4) at *j*th egress interface of *p*th MR. This array has 4 elements for 4 parameter values (i.e. NO_OF_P) corresponding to each egress interface of *p*th MR. The element values of the array corresponding to Delay is initialized by 0.0 ms, the element values of the array corresponding to Unused bandwidth is initialized by the available bandwidth at *j*th egress interface in kbps, the element values of the array corresponding to Packet loss is initialized by 0.0% and the element values of the array corresponding to Cost is initialized by 0.0 units when the system starts functioning. LFN updates the element values of the MR_EGRESS_INFORM data structure after receiving MR_LFN message. LFN updates the value of the 4 array elements (i.e. NO_OF_P) MR_EGRESS_INFORM[p][X1][Y1], MR_EGRESS_INFORM[p][X1][Y2], MR_EGRESS_INFORM[p][X1][Y3] and MR_EGRESS_INFORM[p][X1][Y4] corresponding to the egress interface X1 of the *p*th MR using the values gp(X1,Y1), gp(X1,Y2), gp(X1,Y3) and gp(X1,Y4), respectively. The values corresponding to gp(X1,Y1), gp(X1,Y2), gp(X1,Y3) and gp(X1,Y4) are obtained from *0*th element of the variable array as specified in part3 of the MR_LFN message by *p*th MR. In the best case the number of elements in the variable array is 1 and the element has (NO_OF_P + 1) number of components as discussed in Sect. 2.1.4. So the computation complexity for the updation of data structure 2 is O(NO_OF_P). In the worst case the number of elements in the variable array is 3 (i.e. NO_OF_ST) one per service type and each element has (NO_OF_P + 1) number of components as discussed in Sect. 2.1.4. So the computation complexity for the updation of data structure 2 is O(NO_OF_ST*NO_OF_P). LFN repeats the same steps of updation for the egress interface X2 and X3 of *p*th MR.

The data structure 3 is BE_EG. Each element of BE_EG[p][Z] indicates the best egress of *p*th MR for *Z*th service type. So it is a 2-D array of dimension NO_OF_MR * NO_OF_ST. All the elements of this array are initialized by 1 when the system starts functioning. LFN updates the element values of the BE_EG data structure after receiving MR_LFN message from *p*th MR. LFN updates the values of the 3 array elements (i.e. NO_OF_ST) one per service type, BE_EG[p][Z1], BE_EG[p][Z2] and BE_EG[p][Z3] using X1, X2 and X3, respectively with computation complexity O(NO_OF_ST). The value corresponding to X1, X2 and X3 are obtained from *0*th element, *1*th element and *2*th element of the variable array in part3 of the MR_LFN message as discussed in Sect. 2.1.4.

The data structure 4 is MR_Route_relation. The value of the element MR_Route_relation[r_i][MRp] is 1 if *p*th MR is associated with r_ith route. So it is a 2-D array of dimension NO_OF_R * NO_OF_MR. The value of the elements in this data structure is constant as shown in Table 8.

2.3.2 Generation and updation of routing tables

LFN generates NO_OF_R number of routing tables for NO_OF_R number of routes when the system starts functioning. The number of rows in each routing table is equal to the number of service type (NO_OF_ST). The *i*th routing table has one column to specify the service type, l_i-1 number of columns to specify the best egress interface of l_i-1 number of MRs associated with *i*th route, one column to specify the value attribute for *i*th route and *Z*th service type ($VALUE_i(Z)$). LFN generates NO_OF_R number of routing tables with computation complexity O(NO_OF_R*NO_OF_ST*l_i) as each routing table has NO_OF_ST number of entries each having $l_i + 1$ number of attributes.

LFN inserts the value of the best egress interface in the l_i-1 number of column corresponding to l_i-1 number of MRs associated with the *i*th route in the routing table r_i after receiving MR_LFN message. Let LFN receives MR_LFN message from MRp where $1 \leq p \leq$NO_OF_MR. LFN searches the MR_Route_relation data structure to find the route r_i where $1 \leq i \leq$NO_OF_R with which MRp is associated. LFN inserts BE_EG[MRp][Z1], BE_EG[MRp][Z2] and BE_EG[MRP][Z3] in r_ith routing table. In the best case MRp is associated with a single route and so insertion is required in the 3 rows (i.e. NO_OF_ST) corresponding to the attribute MRp in the r_ith routing table with computation complexity O(NO_OF_ST). In the worst case MRp is associated with NO_OF_R number of routes and so insertion is required in the 3 rows (i.e. NO_OF_ST) corresponding to the attribute MRp in NO_OF_R number of routing tables with computation complexity O(NO_OF_R*NO_OF_ST). For example, let LFN receives MR_LFN message from MR4. LFN searches the MR_Route_relation data structure where it finds MR_Route_relation[r2][MR4] and MR_Route_relation[r4][MR4] is 1. So MR4 is associated with route r2 and route r4. LFN inserts BE_EG[MR4][Z1], BE_EG[MR4][Z2] and BE_EG[MR4][Z3] in Table 5 and Table 7 corresponding to the route r2 and route r4, respectively with computation complexity O(2*NO_OF_ST) as insertion is required in 2 number of routing tables.

LFN computes the value attribute of the routing table after receiving MNN_LFN message. The computation of the value attribute for the service type Z1 and for routes r1, r2, r3, r4 are discussed below. LFN repeats the same steps of operation to compute the value attribute for the services Z2, Z3 and for routes r1, r2, r3, r4.

Let us consider the routing table (Table 4) for route r1. The route r1 is associated with MR1 and MR2. LFN computes $VALUE_{r1}(Z1)$ which is the value attribute for route r1 and service type Z1 using the min operation between MR_EG_STA[MR1][BE_EG[MR1][Z1]][Z1] and MR_EG_STA[MR2][BE_EG[MR2][Z1]][Z1].

Let us consider the routing table (Table 5) for route r2. The route r2 is associated with MR3 and MR4. LFN computes $VALUE_{r2}(Z1)$ which is the value attribute for route r2 and service type Z1 using the min operation between MR_EG_STA[MR3][BE_EG[MR3][Z1]][Z1] and MR_EG_STA[MR4][BE_EG[MR4][Z1]][Z1].

Let us consider the routing table (Table 6) for route r3. The route r3 is associated with MR5 and MR6. LFN computes $VALUE_{r3}(Z1)$ which is the value attribute for route r3 and service type Z1 using the min operation between MR_EG_STA[MR5][BE_EG[MR5][Z1]][Z1] and MR_EG_STA[MR6][BE_EG[MR6][Z1]][Z1].

Table 4

Services	MR1	MR2	$VALUE_{r1}(Z)$
Z1	BE_EG[MR1][Z1]	BE_EG[MR2][Z1]	$VALUE_{r1}(Z1)$
Z2	BE_EG[MR1][Z2]	BE_EG[MR2][Z2]	$VALUE_{r1}(Z2)$
Z3	BE_EG[MR1][Z3]	BE_EG[MR2][Z3]	$VALUE_{r1}(Z3)$

Table 5

Services	MR3	MR4	$VALUE_{r2}(Z)$
Z1	BE_EG[MR3][Z1]	BE_EG[MR4][Z1]	$VALUE_{r2}(Z1)$
Z2	BE_EG[MR3][Z2]	BE_EG[MR4][Z2]	$VALUE_{r2}(Z2)$
Z3	BE_EG[MR3][Z3]	BE_EG[MR4][Z3]	$VALUE_{r2}(Z3)$

Table 6

Services	MR5	MR6	VALUE $_{r3}(Z1)$
Z1	BE_EG[MR5][Z1]	BE_EG[MR6][Z1]	$VALUE_{r3}(Z1)$
Z2	BE_EG[MR5][Z2]	BE_EG[MR6][Z2]	$VALUE_{r3}(Z2)$
Z3	BE_EG[MR5][Z3]	BE_EG[MR6][Z3]	$VALUE_{r3}(Z3)$

Let us consider the routing table (Table 7) for route r4. The route r4 is associated with MR1, MR3 and MR4. LFN computes $VALUE_{r4}(Z1)$ which is the value attribute for route r4 and service type Z1 using the min operation between MR_EG_STA[MR1][BE_EG[MR1][Z1]][Z1], MR_EG_STA[MR3][BE_EG[MR3][Z1]][Z1], MR_EG_STA [MR4][BE_EG[MR4][Z1]][Z1] (Table 8).

The computation of each value attribute needs (l_i-1) number of min operation and so has computation complexity O(l_i-1).

2.3.3 Best route selection algorithm

LFN in the MN uses this algorithm to determine the best possible route per service type supported by MN using the value attribute information available in the routing table. The algorithm selects route r1 for the service type Z1 if $VALUE_{r1}(Z1) \vee VALUE_{r2}(Z1) \vee VALUE_{r3}(Z1) \vee VALUE_{r4}(Z1) = VALUE_{r1}(Z1)$, selects route r2 for the service type Z1 if $VALUE_{r1}(Z1) \vee VALUE_{r2}(Z1) \vee VALUE_{r3}(Z1) \vee VALUE_{r4}(Z1) = VALUE_{r2}(Z1)$, selects route r3 for the service type Z1 if $VALUE_{r1}(Z1) \vee VALUE_{r2}(Z1) \vee VALUE_{r3}(Z1) \vee VALUE_{r4}(Z1) = VALUE_{r3}(Z1)$ and selects route r4 for the service type Z1 if $VALUE_{r1}(Z1) \vee VALUE_{r2}(Z1) \vee VALUE_{r3}(Z1) \vee VALUE_{r4}(Z1) = VALUE_{r4}(Z1)$

Each route selection needs (NO_OF_R-1) number of max operation and so has computation complexity O(NO_OF_R).

If the two different routes are selected as the best possible route for *Z*th service type, the algorithm uses "Best possible route selection function" to select the optimal route.

Table 8 MR_Route_relation data structure

Route	MR1	MR2	MR3	MR4	MR5	MR6
r1	1	1	0	0	0	0
r2	0	0	1	1	0	0
r3	0	0	0	0	1	1
r4	1	0	1	1	0	0

2.3.3.1 Best possible route selection function Let the *d*th and *t*th route (rd and rt) are selected as the best possible route for *Z*th service type where ($1 \leq d \leq$ NO_OF_R, $1 \leq t \leq$ NO_OF_R and $d \neq t$). LFN computes the average delay, average unused bandwidth, average packet loss and average cost of the *d*th route and *t*th route to select the best route for *Z*th service type (BestRoute(Z)). The expression to compute the average delay of *i*th route (avg_delay_ri), the average unused bandwidth of *i*th route (avg_unused_BW_ri), the average packet loss of *i*th route (avg_packet_loss_ri) and the average cost of *i*th route (avg_cost_ri) are as given below:

$$\text{avg\_delay\_ri} = \sum_{p=1}^{li-1} \text{MR\_EGRESS\_INFORM}[p][\text{BE\_EG}[p][Z]][Y1]/(l_i - 1)$$

$$\text{avg\_unused\_BW\_ri} = \sum_{p=1}^{li-1} \text{MR\_EGRESS\_INFORM}[p][\text{BE\_EG}[p][Z]][Y2]/(l_i - 1)$$

$$\text{avg\_packet\_loss\_ri} = \sum_{p=1}^{li-1} \text{MR\_EGRESS\_INFORM}[p][\text{BE\_EG}[p][Z]][Y3]/(l_i - 1)$$

$$\text{avg\_cost\_ri} = \sum_{p=1}^{li-1} \text{MR\_EGRESS\_INFORM}[p][\text{BE\_EG}[p][Z]][Y4]/(l_i - 1)$$

Each expression is evaluated using (l_i-2) number of addition and 1 division with computation complexity O(l_i).

The best possible route selection function uses the expression of avg_delay_ri, avg_unused_BW_ri, avg_packet_loss_ri and avg_cost_ri to compute the average delay, average unused bandwidth, average packet loss and average cost for *d*th route ($i = $ d) and for *t*th route ($i = $ t). In the worst case all the NO_OF_R number of routes is selected

Table 7

Services	MR1	MR3	MR4	$VALUE_{r4}(Z)$
Z1	BE_EG[MR1][Z1]	BE_EG[MR3][Z1]	BE_EG[MR4][Z1]	$VALUE_{r4}(Z1)$
Z2	BE_EG[MR1][Z2]	BE_EG[MR3][Z2]	BE_EG[MR4][Z2]	$VALUE_{r4}(Z2)$
Z3	BE_EG[MR1][Z3]	BE_EG[MR3][Z3]	BE_EG[MR4][Z3]	$VALUE_{r4}(Z3)$

as optimal route for *Z*th service type. In such a case the algorithm uses best possible route selection function for (NO_OF_R-1) number of times to select the optimal route for *Z*th service type with computation complexity O(NO_OF_R*l_i). The avg_delay_rd, avg_unused_BW_rd, avg_packet_loss_rd and avg_cost_rd are the average delay, average unused bandwidth, average packet loss and average cost of *d*th route respectively. The avg_delay_rt, avg_unused_BW_rt, avg_packet_loss_rt and avg_cost_rt are the average delay, average unused bandwidth, average packet loss and average cost of tth route respectively. The steps of operation of the proposed function are given below:

```
if (Z=Z1)
{
 if (avg_unused_BW_rd=avg_unused_BW_rt)
  {
  if (avg_packet_loss_rd = avg_packet_loss_rt)
  {
    if (avg_delay_rd = avg_delay_rt)
    {
     if (avg_cost_rd=avg_cost_rt)
       randomly select one of rd, rt as BestRoute(Z)
     else
     {
       if  (avg_cost_rd<avg_cost_rt)
         BestRoute(Z)<-rd
       else
         BestRoute(Z)<-rt
     }
    }
  else
    {
     if (avg_delay_rd<avg_delay_rt)
         BestRoute(Z)<-rd
     else
         BestRoute(Z)<-rt
    }
   }
   else
        {
        if (avg_packet_loss_rd<avg_packet_loss_rt)
           BestRoute(Z)<-rd
        else
           BestRoute(Z)<-rt
        }
     }
   else
     {
      if (avg_unused_BW_rd<avg_unused_BW_rt)
           BestRoute(Z)<-rt
      else
           BestRoute(Z)<-rd
     }
  }
  else
   if (Z=Z2)
  {
   if (avg_delay_rd=avg_delay_rt)
    {
    if (avg_packet_loss_rd = avg_packet loss_rt)
     {
      if (avg_unused_BW_rd = avg_unused_BW_rt)
      {
       if (avg_cost_rd=avg_cost_rt)
         randomly select one of rd, rt as BestRoute(Z)
       else
       {
         if  (avg_cost_rd<avg_cost_rt)
           BestRoute(Z)<-rd
         else
           BestRoute(Z)<-rt
       }
      }
    else
     {
      if (avg_unused_BW_rd<avg_unused_BW_rt)
         BestRoute(Z)<-rt
      else
         BestRoute(Z)<-rd
     }
    }
   else
     {
      if (avg_packet_loss_rd<avg_packet_loss_rt)
        BestRoute(Z)<-rd
      else
        BestRoute(Z)<-rt
      }
    }
  else
    {
     if (avg_delay_rd<avg_delay_rt)
        BestRoute(Z)<-rd
     else
        BestRoute(Z)<-rt
    }
  }
else
 if (Z=Z3)
{
 if (avg_packet_loss_rd=avg_packet_loss_rt)
  {
   if (avg_delay_rd = avg_delay_rt)
   {
    if (avg_unused_BW_rd = avg_unused_BW_rt)
    {if (avg_cost_rd=avg_cost_rt)
      randomly select one of rd, rt as BestRoute(Z)
     else
      {
       if  (avg_cost_rd<avg_cost_rt)
         BestRoute(Z)<-rd
       else
         BestRoute(Z)<-rt
      }
     }
   else
    {
     if (avg_unused_BW_rd<avg_unused_BW_rt)
        BestRoute(Z)<-rt
     else
        BestRoute(Z)<-rd
    }
   }
  else
    {
     if (avg_delay_rd<avg_delay_rt)
        BestRoute(Z)<-rd
     else
        BestRoute(Z)<-rt
    }
  }
```

```
else
 {
 if (avg_packet_loss_rd<avg_packet_loss_rt)
     BestRoute(Z)<-rd
 else
     BestRoute(Z)<-rt
 }
}
```

LFN uses the same procedure to select the optimal route for the service types Z2 and Z3.

3 Computation complexity of the proposed algorithm

In this section the computation complexity of the best egress determination algorithm and best route selection algorithm to select a route for *Z*th service type is considered for discussion.

3.1 Computation complexity of the best egress determination algorithm

It is the sum of the computation complexity of g(X, Y) calculation, computation complexity of F1 and F2 fuzzifier, computation complexity of R(X, Z) calculation, computation complexity of executing best egress determination algorithm and computation complexity of generating MR_LFN message.

The computation complexity to compute g(j, Y1), g(j, Y2), g(j, Y3) and g(j, Y4) for *j*th egress interface are O(1), O(1), O(number of packets in egress buffer) and O(1) respectively as discussed in Sect. 2.2.2. So the computation complexity of g(X, Y) calculation per egress interface of a MR is O(number of packets in egress buffer) and for NO_OF_EI number of egress interfaces of a MR are O(NO_OF_EI*number of packets in egress buffer).

The computation complexity to compute $F1\_de_{Ej}$, $F1\_BW_{Ej}$, $F1\_PL_{Ej}$ and $F1\_cost_{Ej}$ for *j*th egress interface by F1 fuzzifier are O(NO_OF_EI), O(1), O(1) and O(NO_OF_EI) respectively as discussed in Sect. 2.2.2. So the computation complexity of F1 fuzzifier per egress interface of a MR is O(NO_OF_EI) and for NO_OF_EI number of egress interfaces of a MR are O(NO_OF_EI*NO_OF_EI).

The computation complexity to compute $F2\_De_Z$, $F2\_BW_Z$, $F2\_PL_Z$ and $F2\_cost_Z$ for *Z*th service type by F2 fuzzifier is O(1) as discussed in Sect. 2.2.3.

Each MR computes R(X, Z) by using max-min composition among the fuzzy relations P(X,Y) and Q(Y,Z). Each element of R(X, Z) is computed with computation complexity O(2*NO_OF_P-1) as discussed in Sect. 2.2.4. R(X,Z) has (NO_OF_EI*NO_OF_ST) number of elements. So all the elements of R(X, Z) are computed with computation complexity O(NO_OF_EI*NO_OF_ST*NO_OF_P).

The best egress determination algorithm of a MR has computation complexity O(NO_OF_EI) to select a single egress interface for a service type. If multiple egress interfaces are selected as the best egress interface for a service type, the algorithm uses the best possible egress interface selection function with computation complexity O(number of best egress * NO_OF_P) to select the optimal egress interface as discussed in Sect. 2.2.5.

The MR_LFN message generation has computation complexity O(NO_OF_P) in the best case and O(NO_OF_ST * NO_OF_P) in the worst case as discussed in Sect. 2.1.4.

So the computation complexity of the best egress determination algorithm is O(NO_OF_EI*NO_OF_EI) + O(NO_OF_EI*NO_OF_ST*NO_OF_P) + O(NO_OF_EI* number of packets in egress buffer) + O(number of best egress*NO_OF_P) + O(NO_OF_ST*NO_OF_P).

3.2 Computation complexity of the best route selection algorithm

It is the sum of the computation complexity of data structure updation, computation complexity of the routing table updation, computation complexity of exe cuting best route selection algorithm and computation complexity of generating LFN_MNN message.

LFN updates data structure 1 and data structure 3 with computation complexity O(NO_OF_ST) whereas updates data structure 2 with computation complexity O(NO_OF_P) in the best case and O(NO_OF_ST*NO_OF_P) in the worst case as discussed in Sect. 2.3.1. The computation complexity to search an element from data structure 4 is O(1).

LFN inserts the value of the best egress interface in the *i*th routing table corresponding to the *i*th route after receiving MR_LFN message with computation complexity O((l_i-1)*NO_OF_ST). So the computation complexity for insertion in NO_OF_R number of routing tables is $\sum_{i=1}^{NO\_OF\_R} 0$ ((l_i-1)*NO_OF_ST). LFN computes NO_OF_R number of value attributes for each service type after receiving MNN_LFN message with computation complexity O(NO_OF_R*l_i) as discussed in Sect. 2.3.2.

The best route selection algorithm has computation complexity O(NO_OF_R) to select a single route. If multiple routes are selected as the best route for a service type, the algorithm uses best possible route selection function with computation complexity O(number of best route*l_i) as discussed in Sect. 2.3.3.

LFN generates LFN_MNN message with computation complexity O(l_i).

So the computation complexity of the best route selection algorithm is O(l_i * NO_OF_ST * NO_OF_R) +

O(NO_OF_R * l_i) + O(NO_OF_ST * NO_OF_P) + O (number of best route * l_i).

4 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 a 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 software. NEMO_SIM can be a part of NS2 simulation environment by using AgentJ [11], which is a JAVA Virtual Machine for NS2. NEMO_SIM can also act as an extended part of JNS 1.7, JAVA Network Simulator [12]. 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 processing units and the corresponding threads are shown in Fig. 5. The function of all the threads is discussed in the following sections.

4.1 MNN_REQ (T_1) thread

It sends MNN_LFN message to LFN. A MNN has only one T_1 thread.

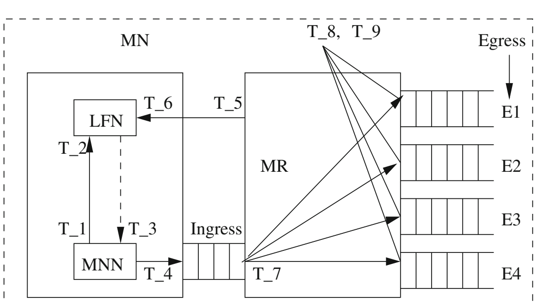

Fig. 5 Threads

4.2 LFN_MNN (T_2) 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 (T_3) thread

It receives LFN_MNN response message and starts a new session. A MNN has only one T_3 thread.

4.4 MNN_SERVICE (T_4) 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 corresponding to the optimal route. After transmitting all the packets successfully T_4 thread dies. A MNN has zero or more T_4 thread depending upon how many sessions are still alive.

4.5 MR_ROUTE_UPDATION (T_5) thread

It performs the operation of F1 fuzzifier and sends MR_LFN message.

4.6 LFN_MR (T_6) thread

It receives MR_LFN message.

4.7 MR_PACKET_RECEIVE_FORWARD (T_7) thread

It receives a packet from the ingress queue and forwards it to the best egress as selected by the phase 1 of the route selection algorithm.

4.8 MR_egress (T_8) thread

It receives a packet from the best egress queue and forwards it to the ingress queue of the next hop as specified in routing table. It also computes packet loss due to the overflow at the egress queue.

4.9 MR_EGRESS_PACKET_LOSS (T_9) thread

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

5 Simulation

The simulation experiment is carried out for 3 different cases considering the internal network of NEMO (Fig. 1) as WiFi (IEEE 802.11a). Each case of experiment considers different size of LFN buffer, MR egress and ingress buffer and MNN buffer as mentioned below:

Case I: LFN buffer size 1000, MR egress and ingress interface buffer size 10^5, MNN buffer 1000
Case II: LFN buffer size 1500, MR egress and ingress interface buffer size 150000, MNN buffer 1500
Case III: LFN buffer size 500, MR egress and ingress interface buffer size 50000, MNN buffer 500

The performance of the proposed NEMO is studied using NEMO_SIM simulator.

Figures 6, 7 and 8 show the plot of throughput, route selection time and session loss vs. traffic load of the proposed NEMO (Fig. 1) for 3 cases. The traffic load is computed as the ratio of arrival rate and departure rate of session request from MNN by LFN. In case I route selection time is constant up to traffic load 300 and then it increases slowly which causes decrease in throughput with traffic load. Session loss is zero up to traffic load 350 and then it increases slowly. In case II route selection time is constant up to traffic load 550 and then it increases slowly which causes the decrease in throughput with traffic load. Session loss is zero up to traffic load 550 then it increases slowly with traffic load. In case III route selection time is constant up to traffic load 200 and then it increases slowly which causes the decrease in throughput with traffic load. Session loss is zero up to traffic load 350 then it increases slowly with traffic load.

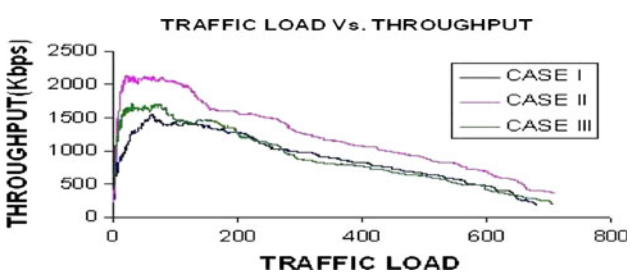

Fig. 6 Throughput vs. Traffic load for NEMO in Fig. 1

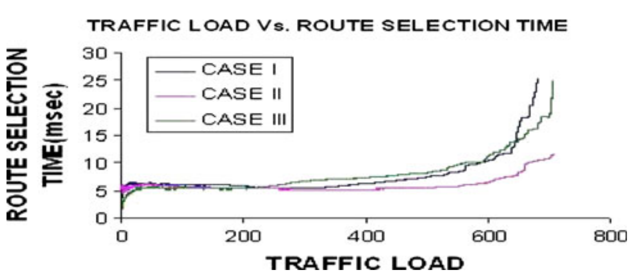

Fig. 7 Route selection time vs. Traffic load for NEMO in Fig. 1

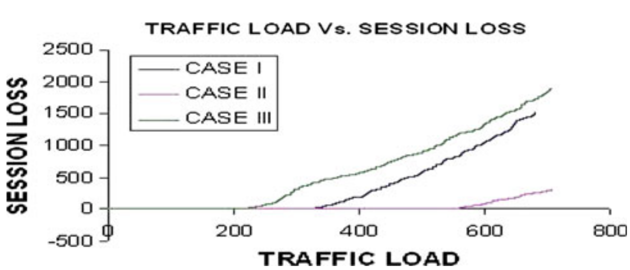

Fig. 8 Session loss vs. Traffic load for NEMO in Fig. 1

The route selection time is minimum, throughput is maximum and session loss is minimum in case II due to maximum buffer. The route selection time and session loss is maximum in case III due to minimum buffer. The route selection time and session loss in case I is higher than case II but lower than case III due to moderate buffer. The route selection time in case III is lesser than the route selection time in case I up to traffic load 200 which causes higher throughput in case III than in case I. The route selection time in case III is higher than the route selection time in case I from traffic load 200 which causes lesser throughput in case III than in case I.

6 Comparison of the proposed work with existing schemes

Deleplace and Noel proposed route optimization in nested mobile networks [13]. It is a policy based route optimization algorithm. The route selection policy in this algorithm is to select the router that is linked with the root MR that provides the best bandwidth if it's nested level is maximum. It also advices to select less nested path as the best path and also to select higher throughput path. But in the proposed work the best route is selected based on the metrics like delay, available bandwidth, packet loss and cost. A less nested route may not always remain the best route. The proposed work also concerns about service type specific route selection. It takes care whether a service is loss sensitive, delay sensitive, requires high bandwidth or costly.

Clausen and Baccelli LIX proposed a route optimization scheme [14] in nested mobile networks using optimized link state routing [15] protocol. But this scheme is mainly designed for ad-hoc wireless networks. It does not sense the quality of the link. It assumes that a link is up if a number of hello packets have been received recently. It does a lot of flooding which consumes a large bandwidth and CPU power to compute optimal paths. In the proposed work flooding is not required which helps to reduce the CPU power consumption.

7 Conclusion

This paper has presented the use of fuzzy logic concept for route optimization in multihomed mobile networks. Each

MR determines its best egress interface per service type using best egress determination algorithm and sends this information to the LFN inside the MN. LFN uses this information for executing the best route selection algorithm and selects an optimal route for the desired service type of MNN. The simulation result shows the efficiency of the proposed scheme in terms of throughput, route selection time and session loss. The proposed scheme can be extended to provide communication between MNN in MN and 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. Perkins, C. (2002). "IP mobility support for IPv4", IETF RFC 3344, August.
2. Johnson, D., Perkins, C., & Arkko, J. (2004). "Mobility support in IPv6", IETF RFC 3775, June.
3. Devarapalli, V., Wakikawa, R., Petrescu, A., & Thubert, P. (2005). Network mobility basic support protocol, IETF RFC 3963, January.
4. Cho, S., Na, J., & Kim, C. (2005). A dynamic load sharing mechanism in multihomed mobile networks, pp. 1459–1463, ICC.
5. Wakikawa, R., Devarapalli, V., Thubert, P. (2004). "Inter home agents protocol (HAHA)", IETF internet draft, draft-wakikawa-mip6-nemo-haha-01, February.
6. Shima, K., Uo, Y., Ogashiwa, N., Uda, S. (2006). Operational experiment of seamless handover of a mobile router using multiple care-of address registration, *Journal of Networks, 1*(3).
7. Wakikawa, R., Ernst, T., Nagami, K. (2006). "Multiple care-of adresses registration", IETF, technical report draft-wakikawa-mobileip-multiplecoa-05, February.
8. Adeniji, S.D., Khatun, S., Raja, R.S.A., Borhan, M.A. (2008). "Design and analysis of resource management support software for multihoming in vehicle of IPv6 Network", pp.13–17, AsiaCSN.
9. Ernst, T., Charbon, J. (2004). "Multihoming with NEMO basic support", ICMU, Yokosuka, Japan, January.
10. www.3gpp.org/FTP/tsg-sa/WG1.Serv/TSGS1_03-HCourt/Docs/Docs/S1-99362.docTSG-SA Workinggroup1(services) meeting#3, hampton court, surrey, UK 10th–12th May, 1999.
11. Taylor, I., Downard, I., Adamson, B., Macker, J. (2006). 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.
12. Java network simulator: http://jns.sourceforge.net.
13. DELEPLACE, A., Noel, T. Workshop NEMO 2007 multihoming in nested mobile networks with route optimization http://www.nautilus6.org/events/0701-WONEMO/20070115-WONEMONestedMultihoming.pdf.
14. Clausen, T., Baccelli LIX, E., Wakikawa, R. "Route Optimization in Nested Mobile Networks (NEMO) using OLSR", http://www.emmanuelbaccelli.org/publications/NCS_2005.pdf.
15. http://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol.

Author Biographies

Sulata Mitra received B.E. degree from Bengal Engineering College (India) in 1986 and Ph.D. degree in Mobile Computing from Bengal Engineering and Science University, Shibpur (India) in 2005. She joined the Indian Institute of Technology, Kharagpur in 1989 as Senior Research Assistant and moved to the Regional Institute of Technology, Jamshedpur (India) in 1991 as Lecturer. Dr. Mitra has published 32 technical papers in journal and international conference proceedings. Her current research interest is QoS-issues in 3G/4G cellular network. She is currently with the Computer Science and Technology department of Bengal Engineering and Science University, Shibpur (India) as Assistant Professor.

Sumanta Pyne B.Tech. graduated from the Department of Computer Science and Engineering from Meghnad Saha Institute of Technology, Kolkata, India in 2005. He started his career as a programmer at HQ Solutions, Kolkata. Then he joined Techno India Institute of Technology, Kolkata as a lecturer at the Department of Computer Science and Engineering. He has done Master of Engineering from the Department of Computer Science and Technology, Bengal Engineering and Science University, Shibpur (formerly Bengal Engineering College), Howrah, India. He is presently doing his PhD in the Department of Computer Science and Technology at IIT Kharagpur.