

## **A novel host address auto-allocation based on moose MAC address**

Yu-Bing Mao and Zheng-You Xia

*College of Computer Science and Technology, Nanjing University of Aeronautics and  
Astronautics, Nanjing City, 210016, China  
E-mail: zhengyou\_xia@nuaa.edu.cn*

The introduction of virtualization and the expansion of scale in data center result in sharp consumption of MAC address, which also expose some weakness in MAC address such as inefficient allocation scheme, and one-dimensional address structure, among others. Varieties of solutions are proposed by researchers who attempt to solve above limitation in traditional MAC address. However, none of those approaches is optimum for hosts' MOOSE ID distribution. In this paper, we present a host address assignment scheme termed Host Moose Address Auto-allocation (HMAA) based on Moose Mac address. Host active sends initiate the application of Moose Mac address allocation to the home switch in our scheme. We make some experimental simulations to study performance of our scheme, and the simulation result shows that our scheme is feasible and effective.

*Keywords:* MAC Address; Moose Address; OPNET; Host Address Allocation.

### **1. Introduction**

MAC address space will be exhausted in the near future. Nowadays, EUI-48 identifier not only is used as an identifier of network or media access control (MAC) address service to locate physical device what IEEE project 802 designed MAC address to do, but also identifies protocols and other functions[1]. However, each physical device can virtual multiple devices, that is, each physical device has various MAC addresses when the virtualization is introduced to tackle the issues of low resources utilization rate caused by rapid information and business expansion demand in data center. Hence, this lead to explosive growth of MAC addresses space. Moreover, the lifetime of MAC address is always permanent or the whole life cycle of equipment. Accordingly, short lifecycle equipment, such as Thermostat and light sensor, occupy MAC address space rapidly. Last but not least, extensive usage of IoT devices also aggravates MAC address consumption. Consequently, MAC address can be rapidly consumed. Besides, MAC address itself also has some problems. MAC address is one-dimensional and can't provide device location which lead to the

requirement of switch to discover and store every addressable devices in the network[2]. However, content addressable memory (CAM) storing forwarding tables doesn't match growth rate of table length [3, 4] which can cause broadcast storm once data overflow. Therefore, virtual machine can only be migrated in layer 2 to maintain the continuity of virtual machine bearer service. Whereas, data center restrict the size of broadcast area to avoid broadcast storm caused by STP protocol [5].

Hence, new-type MAC address is an important task and is a study field needing further exploration. Although these challenges persist, researchers have put forward some solutions to the problems mentioned above [6, 7, 8, 9 ]. Among the numerous MAC address structures, MOOSE address [9] is associated three bytes host identifier with fixed three bytes switch identifier. In our previous work, we proposed a centralized management address allocation scheme based on the structure of the MOOSE address aimed at these problems. Through the election of Designated Switch and Backup Designated Switch, we accomplished centralized allocation and management of the switch MOOSE address within the network. However, none of those approaches are optimum for MOOSE ID of hosts' distribution. Hence, we propose a scheme for host address assignment. In this paper, we focus on the address allocation of hosts and propose a host address assignment scheme termed Host Moose Address Auto-allocation (HMAA) to reduce edge switches overhead. Computer simulation is conducted to evaluate the performances of address allocations including address collision rate, time, etc. The simulation results show our previous scheme is feasible.

## **2. Our Scheme**

Our solution is based on a centralized management switch address allocation scheme in our previous work. Designated Switch (DS) and Backup Designated Switch (BDS) need to be elected and the DS allocate MOOSE ID for every switch in the network before host can request address from its direct-connected switch. In this section, we describe our scheme that can reduce burden of edge switches when allocating addresses for hosts which does not require an edge switch changes source address to MOOSE ID every time a host sends packets. And the switch only needs to maintain MAC/MOOSE address mapping table after the edge switch assigns address for the host. In addition, communication with other hosts shall not be attempted until the host address allocation procedure is complete.

Initially, host will send the Host Request (HR) message to the connected switch for host MOOSE ID and start a timer. This message includes the following information: a) source address which can be all zero or expected

MOOSE ID, b) destination address that is specific multicast address, c) event id randomized by the host, d) MAC address of the host. e) MOOSE\_ID that can be all zero or an expected MOOSE address. If the timer expires before receiving any frame, the host will resend the same Host Request message one more time before retransmission trial time reaches the max value (Fig. 1 (b)).

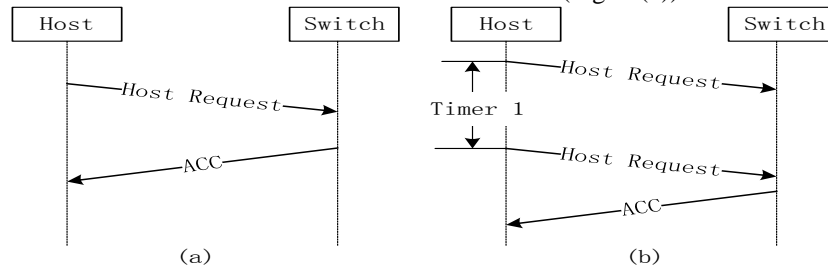


Fig. 1 Reply ACC

Once the switch receive the HR message, it sends RJT frame with corresponding value in Conference field indicating rejection to the request if its own MOOSE Switch ID is not allocated by DS (Figure 2 (b)). Otherwise, the switch estimates if it is too busy to handle other requests. If so, the switch responds with a BSY frame (Figure 3). If not, the switch will assigned address for the host based on MOOSE\_ID field within the frame. If the field is all zero indicating that host address can be auto-allocation, the switch will generate 3 bytes host identifier based on MAC address of the host within the HR frame and associate it with its own MOOSE Switch ID. Next, the switch records the host MOOSE ID into the HOST/MOOSE table and responds with an ACC message (Figure 1 (a)) including the following information: a) event id same as the id in HR message, b) MOOSE\_ID that allocated by the switch. If MOOSE\_ID field in the HR message is a certain value, the switch will determine whether the value is already assigned. If so, the switch replies RJT message (Figure 2 (a)) with appropriate value in Conference field. If not, the switch will associate the MOOSE ID with its own MOOSE Switch ID to produce new host MOOSE ID and reply an ACC message (Figure 1 (a)).

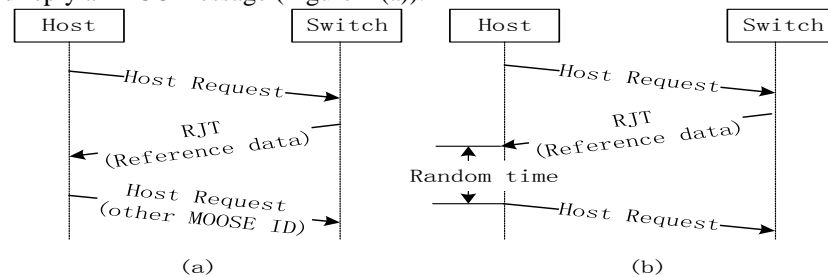


Fig. 2 Reply RJT

Finally, the host receives a frame from its direct-connected switch before the timer expires. Above all, the host checks whether event id in the frame is the same as the id within former HR message sent by itself. If not, the host discards the received message. Or else, the host determines the frame type. If it is a RJT frame and the Conference field indicates the switch is not read to allocate address for host (Figure 2 (b)), the host will wait for some random time and resend the same HR message. If it is a RJT frame and the Conference field indicates requested MOOSE ID is unavailable (Figure 2 (a)), the host will choose another MOOSE ID and request for this new address. If the type of the receiving message is BSY (Figure 3), it indicates the switch is busy and the host will wait for some random time until resending the HR message. If it is an ACC frame (Figure 1 (a)), the host obtain MOOSE address from the MOOSE\_ID within the ACC frame and can communicate with other hosts in the network.

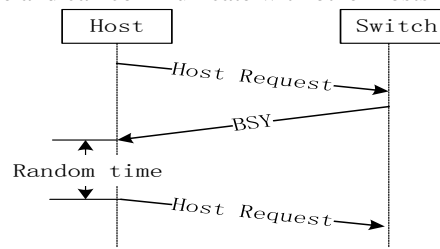


Fig. 3 Reply BSY

### 3. Experimental Result and Discussion

In this section, we evaluate the performance of the proposed host MOOSE address auto-allocation protocol in terms of efficiency and less collision. To this end, we have implemented the solution on the OPNET Modeler 14.5 for simulation.

#### 3.1. Simulation setup

Our solution modifies the structures of Ethernet station model and switch model adopted in the simulation node model for the convenience of host address assignment. And we design Distribution Process to handle the most of assignment (Fig. 4).

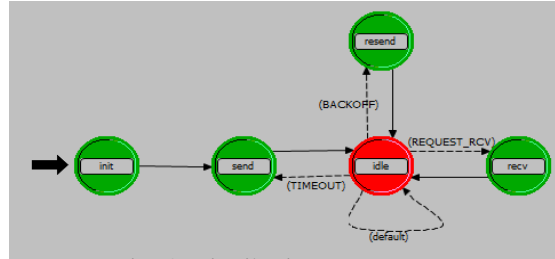


Fig. 4. Distribution Process

In each scenario of the simulation, the source node and the destination node are randomly combined. In order to ensure the whole network in the high load condition, Interarrival time (seconds) for generating packets obey exponential (1.0) and packet size (bytes) obey uniform (46, 1500). The simulation of the reunification of the time set for 1 hour.

### 3.2. Simulation results analysis

Initially, we evaluate the average time of all hosts it will take to receive ACC frame. In Figure 5, we present the results for both the time after host request from its direct-connected switch and after the switch obtain its own MOOSE ID. The results show that the time is relatively stable separately. As the number of hosts increase, the Receive ACC Time changes little. However, the time that switch wait is a considerable factor for host moose address allocation. Subsequently, we evaluate the throughput in the network. Originally, the throughput is low because hosts don't have MOOSE ID. After more host MOOSE IDs have been allocated, the throughput rises rapidly and then remains stable. The results of Figure 6 show that our solution does not have negatively impact of throughput in the network. Next, we evaluate the MOOSE ID collision rate of the proposed protocol. To this end, we confirm that HMAA we describe in this paper won't cause duplication of addresses in the whole network through the results depict in Fig. 7.

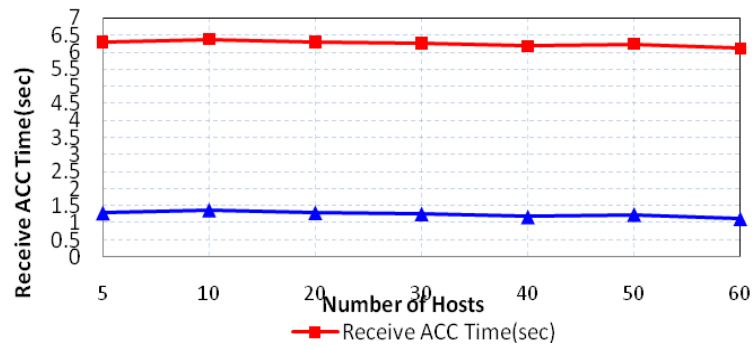


Fig. 5 Receive ACC Time vs. number of hosts

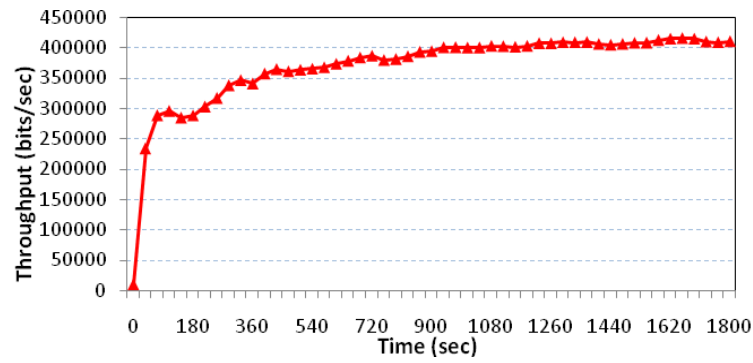


Fig. 6 Throughput vs. Time

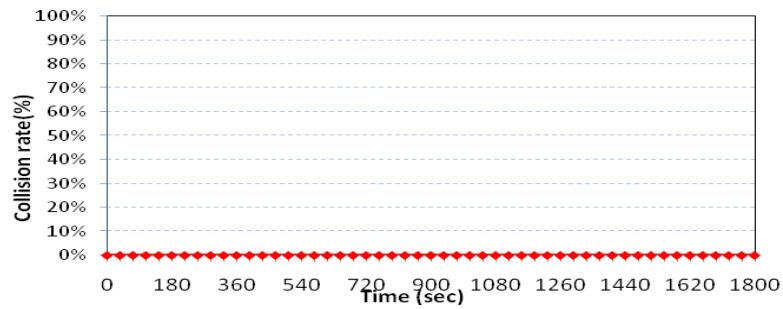


Fig. 7 Collision rate vs. Time

#### 4. Conclusion

In this paper, we have presented a host address assignment scheme termed Host Moose Address Auto-allocation (HMAA) based on our previous work. The proposed protocol reduces edge switches overhead which does not require an edge switch changes source address to MOOSE ID every time a host send packets. And the switch only needs to maintain MAC/MOOSE address mapping

table after the edge switch assigns address for the host. In the meantime, we build the network model, node model and process model of proposed protocol in the simulation as required using OPNET in different topologies. And the proposed protocol is viable, as confirmed by simulation results. Whereas, this solution needs to be perfected in some aspects. In the future work, we plan to investigate and discuss how to allocate and maintain address in live migration of virtual machines.

### Acknowledgments

This paper was supported by the ZTE (ZhongXing) Corporation industrial-academic-research cooperation project.

### References

1. IEEE Std 802-2001. The Institute of Electrical and Electronics Engineers, Inc. (IEEE). 2002-02-07. p. 19. ISBN 0-7381-2941-0. Retrieved 2011-09-08.
2. 3Com Corporation, Switch 5500G 10/100/1000 family data sheet. Information [onhttp://www.3com.com/other/pdfs/products/enUS/400908.pdf](http://www.3com.com/other/pdfs/products/enUS/400908.pdf).
3. F. Yu, et al., Efficient multimatch packet classification and lookup with tcam, IEEE Micro, vol. 25, no. 1, Jan. 2005.
4. K. Pagiamtzis and A. Sheikholeslami, Content Addressable Memory (CAM) circuits and architectures: a tutorial and survey, IEEE Journal of Solid-state Circuits, vol. 41, pp. 712–727, 2006.
5. IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, Sponsor: LAN MAN Standards Committee of the IEEE Computer Society, Approved 4 June 2004(American National Standard Institute), Approved 9 February 2004(IEEE-SA Standards Board).
6. Guideline for use of EUI, IEEE standards association.
7. Dan Harkins, Aruba Network, 802c and the Local Address Space, November 2014.
8. Roger B. Marks, EthAirNet Associates, Making Room for IEEE 802 Protocols in the Local Space, 2014-11-05.
9. Malcolm Scott, Daniel Wagner-Hall, Andrew Moore and Jon Scowcroft, Addressing the Scalability of Ethernet with MOOSE, University of Cambridge Computer Laboratory.