International Journal of Security and Its Applications
Vol. 10, No. 2 (2016), pp.423-436
http://dx.doi.org/10.14257/ijsia.2016.10.2.36

Secure Linear Feedback Shift Register based IPv6 in VANET

Abhishek singh[1], Shekhar Verma[2] and Geetam Singh Tomar

[1,2]*Department of WCC, Indian Institute of Information Technology Allahabad, India*
Machine intelligence Research Labs, Gwalior 474011
singh.0810@gmail.com, sverma@iiita.ac.in, gstomar@ieee.org

Abstract

In this paper, Linear Feedback Shift Register (LFSR) based IPv6 address generation has been proposed for low IP acquisition time during handoff in Vehicular Adhoc Networks (VANET). The mechanism reduces handoff latency and packet loss rate by removing link layer address conflict and duplicate address detection (DAD). VANETs are characteristics by frequent handoffs that need regular address reconfiguration which increases the handoff latency. The latency increases further due to the need for DAD to prevent link layer address conflict. LFSR based IPv6 in VANET decreases the address reconfiguration time and handoff latency by providing a unique seed value in Router Advertisement (RA) which is generated through modified LFSR algorithm and is combined with vehicle's MAC ID to give a unique host ID. Since the mechanism is able to generate different unique host ID for same MAC ID, it eliminates the need of DAD by minimizing the probability of link layer address conflict to a negligible value. This also increases the IP configuration scalability. The algorithm only modifies the control signal RA and does not require any change in the VANET architecture. Simulation results confirm that this mechanism performs better than Mobile IPv6 (MIPv6) and can be incorporated as a standard IP acquisition technique for high mobility networks like VANET.

Index Terms*— IPv6, Vehicular Adhoc Networks, IP acquisition, Linear Feedback Shift Register*

Introduction

VANET is an infrastructure supported mobile network formed by vehicles communicating in either Vehicle to Infrastructure (V2I) or Vehicle to Vehicle (V2V) mode. In V2I, vehicle attaches itself with Road Side Unit (RSU) for outside world connectivity and in V2V vehicles inside a communication range forms an adhoc network to communicate between each other. A vehicle participates as a node and also behaves as a router. This property of acting as intermediate router helps VANET to grow large which is theoretically infinite. VANET supports almost all infotainment services. As a vehicle moves on the road, it leaves one RSU region and needs to join new RSU to run the services. This necessitates a handoff. In a handoff, a vehicle detaches itself from the existing RSU to associate with a new RSU. Figure 1 shows the steps carried out during handoff: triggering, discovery, authentication, association, IP address acquisition and Home Agent (HA) registration. In the first step, a vehicle checks for the quality of signal. If the signal value is less than a predefined threshold, a handoff is triggered. This is

ISSN: 1738-9976 IJSIA

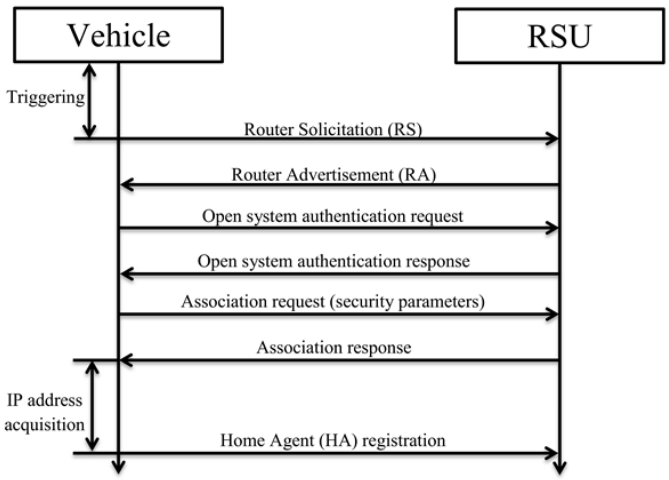


Figure 1. Handoff Steps

followed by discovery in which vehicle searches for new RSU through active or passive scan. In an active scan, probe message is sent on all channels, while in passive scan, a vehicle listens to all channels periodically. After vehicle gets an RA, it authenticates itself through a shared key or open system authentication. VANET uses open system authentication. During the process of association, a vehicle negotiates with the RSU the data rate and reserves resources for itself. Vehicle sends association request consisting of desired Service Set Identification (SSID) and supported data rates. In response, the RSU sends the supported data rate and the session ID. Till this point, the messages require only link layer address for communication but to communicate with outside world, a vehicle needs a global unique IPv6 address which is configured during the IP acquisition process. The Home Agent (HA) registration takes care of updating HA with the newly acquired address. Most of the handoff steps are performed sequentially. However, to minimize the latency some steps are performed concurrently.

Handoff latency is the sum of time taken by each step during handoff process. In this paper we have focused on minimizing the IP acquisition time which constitutes a large fraction of the total handoff time. IP acquisition can result in a delay of 1000 milliseconds which is intolerable for real time applications such as relaying safety messages, VoIP, online banking services, multiplayer gaming etc. A major delay is caused due to IP address conflict, which is common in IPv4 because of smaller pool of address. Dynamic Host Configuration Protocol (DHCP) server [1][2] is used to solve the problem. IPv6 also supports DHCP but is limited by drawbacks like requirement of dedicated resource reservation and increased message overhead due to additional DHCP signaling. A new mechanism of address auto-configuration is proposed in IPv6 which allows a vehicle to configure their IP address automatically based on their MAC address using IEEE EUI64. With a larger pool of address, it works for most of the time. However, in this address auto-configuration mechanism; there is a high probability of an address conflict in link local address. The IP acquisition solutions for traditional fixed networks and mobile ad-hoc networks also cannot be directly applied on VANETs [3][4][5][6][7][8]. In this paper, we propose a new IPv6 acquisition solution based on Linear Feedback Shift Register (LFSR) which introduces randomness in the IP address to minimize the address acquisition time and IP conflicts.

The rest of the paper is divided as follows. Section 0 describes the problem statement, section 0 goes through the previous works done in IP acquisition. Section 0 describes the

proposed algorithm with LFSR mechanism, network architecture, frame format and pseudo code. Section 0 contains the simulation and performance evaluation followed by section 0 which concludes the paper.

System Description and Problem Description

A. VANET handoff Challenges

To provide a moving vehicle with constant connectivity and remain a member of the VANET, handoff is necessary. An RSU region in VANET can cover at most 1 km which is small as compared to a typical cell size. This leads to frequent handoff. To minimize the packet loss, link layer forwarding is combined in VANET handoff. It consists of two units, store and forward and handoff detection. Moreover, since vehicles move on the road at high speeds, the stay time of vehicles in an RSU region is small. During their stay period, vehicles continually exchange messages. Thus, significant packet delay and even packet loss may result during the handoff process. Thus, the handoff overhead should be minimal to avoid delay or losses and also allow to vehicles to exchange messages. While general handoff is focused to terminal mobility, VANET requires network mobility as the MRs (Mobile Router) are mounted on vehicles, The problem gets worse with multiple MR installed vehicles such as bus, train etc. Handoff with multihop communication is not possible in VANETs as performance degrades due to additional hop delay [21].

B. Problem Description

Packet based network needs a unique IP address for communication IP address configuration in wired network is easy and scalable but it is difficult in wireless networks with mobility. The problem of IP configuration gets worse in VANET due to variable nature of node distribution, high speed and dynamic movement. Routing protocols [9][10][11] in VANET lead to either a direct communication between the vehicles or with the help of intermediate vehicle. In both cases, IP address for preparing routing tables and paths.

RSU enables a vehicle to connect with outside world. Even using infrastructure network, vehicle needs their own unique IP to send and receive packets. Most of the infrastructure based networks use DHCP but in a VANET, DHCP cannot be used due to its signaling overhead and dedicated resource requirement. IPv6 address auto-configuration can be a solution but it cannot work with a conflicting link local address and further manual intervention is needed for its working.

The few seconds required by a vehicle to configure their IP address and perform DAD is crucial when it comes to relay an emergency message such as an accident or while running some real time application such as VoIP. It is also important in the case of time sensitive applications such as net banking where delay can lead to session expiry. This delay or latency of few seconds can compromise the performance of a user in an online gaming. Hence, an efficient technique is needed to minimize the delay in IP acquisition without IP address conflicts.

Related Work

IP address acquisition solution proposed in [12][4] are based on decentralized approach. A vehicle needs to request for the address and in reply it gets an address configuration based on its interaction with other nodes. The solution is simple but suffers from high signaling overhead and large configuration time. Different solutions [7][8] work with duplicate addresses until and unless nodes with same address starts communicating with each other. If there are any VANET topologies where some vehicles do not communicate with each other, then these solutions are applicable to minimize the address configuration time. But with the regular VANET topologies, these solutions cannot be implemented directly and require more address reconfigurations leading to higher latency.VAC [13] proposes concept of leaders in the network, which maintains the address list and assigns address to new vehicles. However with high mobility and dynamic topology, the initial leaders cannot be guaranteed to exist throughout the network life time. Merging and splitting of networks leads to new leader elections and dismissal. This again increases the address configuration time and handoff latency. RAPACA [14] proposes solution based on dividing the geographical boundary (MAP) into several small regions based on direction. The region code prefix is determined through its direction from central region. Each region is equipped with a 16 bit unique ID which the vehicle includes in their IP address along with host ID. This 16 bit unique ID informs about its home

region and kept constant irrespective of mobility. This solution suffers from static region partition where adding of removing any AP can lead to repartitioning of the region. Also this solution was concentrated to a particular geographical location, therefore not a suitable global solution. EBFH [15] is a solution that supports early triggering of handoff and address configuration. With high mobility, sometimes the handoff signaling is unable to complete and this leads to dead vehicular node. So to cope up with this problem EBFH proposes to discover the forthcoming AP in advance from the RA and should do the address binding accordingly. This gives extra configuration time to the vehicle. Like other standard methods, this also suffers from signaling overhead and heavy processing for pre-configuration. Solution proposed in MIPv6 for V-WINET/V-ITS [16] calls for preserving the address until the AP forward chain is long enough for intolerable delay. Here vehicle acquired CoA is preserved and in place of reconfiguring IP address with new AP, a forward mechanism is introduced in which the previous AP forwards the packet destined for vehicle to the new AP. The preserved address is termed as oCoA i.e. old CoA and when this AP forward mechanism introduces significant delay, a new address is configured which is called nCoA i.e. new CoA. Similarly forwarding AP and forwarded AP are termed as oAR and nAR respectively. This solution eliminates the need of address reconfiguration but at the cost of forwarding chain and security compromise in terms of ingress filtering.

LFSR based IPv6

In the proposed mechanism, LFSR is used for generating a pseudo random number (seed), which is then operated with MAC address to generate a random host id. Since LFSR does not require high processing and can be integrated into the hardware, it provides the required speed to minimize latency. Also with randomization in IP address, the probability of IP address conflict becomes negligible. The technique modifies the basic LFSR to include a feedback on cycle. This provides added security to the algorithm.

C. Linear Feedback Shift Register (LFSR)

It is a fast method to generate pseudo random numbers. LFSR [22] basically is a shift register whose input bit relies on the output of previous state function. In general, exclusive-or (XOR, ⊕) is performed on combination of register bits which computes the next input bit. The number is then right shifted by 1 bit and input bit is inserted in the MSB position. The initial value of a LFSR is called its seed and since it uses a deterministic function hence the output can be determined from its current or previous state. Due to limited number of states, LFSR cycles after some state but carefully chosen function can keep the states random for a long time.

Fibonacci LFSR has an additional feedback mechanism to make the states more random. In basic Fibonacci LFSR, function keeps the input bit position constant i.e. at the time of function development if the input bit position identified are x, y and z then these bit positions will remain same for all the values. But in the proposed LFSR, the function input bit position keeps on changing depending on the initial seed. We take a 32 bit initial seed from the system clock. Then, by taking the last 4 digits individually, the function input bit positions are determined. These bit positions will remain same as long as cycle does not occur. On the event of cycle, seed is initiated again and same process is repeated. As shown in **Error! Reference source not found.**, the initial seed is 1162933874. Hence the input bit positions for LFSR function will be 4, 7, 8 and 3.The operation performed in the function is given by Equation (1).

$$\text{Input bit} = (((\text{bit1} \oplus \text{bit2}) \oplus \text{bit3}) \oplus \text{bit4}) \wedge 1 \quad (1)$$

Error! Reference source not found. shows the working of LFSR function. Bit positions are marked from 1 to 32 on the top of bit stream. 3, 4, 7 and 8 are the function's input bit positions and are also marked with respective bit position. The value of these bit positions is also underlined. Proposed LFSR functions works in the following manner:

$$\text{Input bit} = (((\text{4th bit} \oplus \text{7th bit}) \oplus \text{8th bit}) \oplus \text{3rd bit}) \wedge 1$$

1 bit right shift operation is performed on the seed and then the above calculated input bit is inserted in the 1st bit place of the seed. This gives the next seed. **Error! Reference source not found.** shows the next state output of the proposed LFSR algorithm. This resulting state serves as input for next state. Steps shown in **Error! Reference source not found.** and **Error! Reference**

source not found. are repeated until a cycle is encountered. After a cycle, step shown in **Error! Reference source not found.** is repeated to give algorithm a fresh start. Initializing seed again on cycle and determining the input bit positions each time from initial seed provides a more random nature to Fibonacci LFSR. Period for each seed in proposed algorithm is different which makes it robust against any kind of attacks.

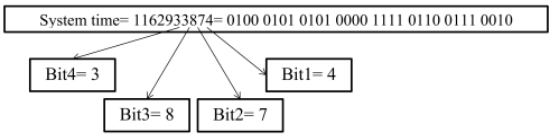


Figure 2. Initial Seed

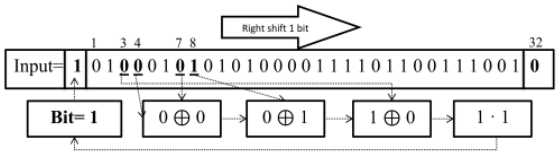


Figure 3. LFSR Function

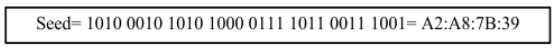


Figure 4. Next State

D. Address Acquisition Mechanism

Error! Reference source not found. shows the network architecture of VANET. It contains various entities such as vehicles, RSU, AR, Internet and Corresponding Node (CN). The dashed circle represents the range of the RSU. RSU acts as point of attachment or access point for the vehicles. Wireless channel is used to communicate between vehicles and RSU. RSUs are connected with AR using wired/wireless channel. AR provides gateway for outside world connection to the RSUs. A CN is any node on Internet wants to communicate with any vehicular node. When vehicle comes into the overlapping RSU region, it checks for signal quality and changes the point of attachment as when required.

When a vehicle goes for handoff, either it actively sends probe i.e. Router Solicitation (RS) or waits for RA from RSU. RA contains various fields as explained in Frame format. In our algorithm, we have introduced a new option along with link layer address, network prefix and MTU which is LFSR 32 bit seed. The first 64 bit of IPv6 address is filled with the desired network prefix extracted from RA's option. For the next 64 bit block we will use the 32 bit seed value. **Error! Reference source not found.** shows the procedure to generate a valid IPv6 address using the32 bit seed. The 32 bit seed encapsulated in RA is left shifted by 32 bits, making it a 64 bit number. Logical OR is performed on this 64 bit and 48 bit MAC ID. This OR result ultimately produces a 64 bit host ID which is combined with network prefix to make a global unique IPv6 address. Since the proposed method uses random state with irregular period, hence this reduces the probability of IP conflict to a very negligible value. In this way LFSR based IPv6 solves the address conflict problem in IPv6 stateless auto-configuration method. With multiple 32 bit seed, a vehicle can even configure itself with multihoming, which is going to be next generation network. It even reduces the IP acquisition delay be discarding the need of DAD. Also by including seed option in RA itself, signaling overhead is kept in control which in return lowers the delay. Seed reset and irregular cycle period makes this algorithm robust against attacks for guessing the next state.

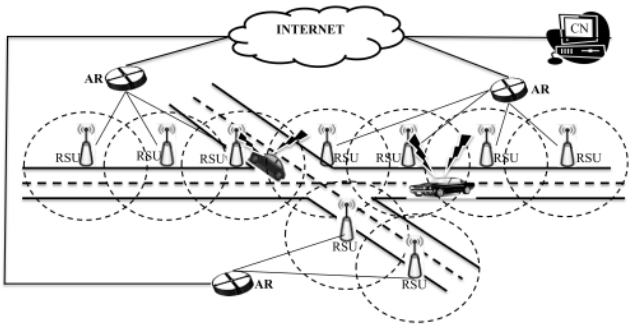


Figure 5. Network Architecture

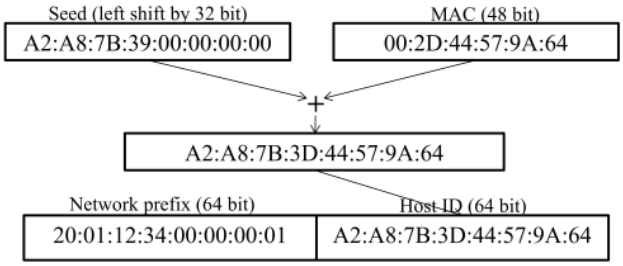


Figure 6. LFSR based IPv6

E. Frame format

A RA consists of various fields as shown in **Error! Reference source not found.**. The main header consists of:

- Type: Signifies the message type. E.g. 128- echo request [17], 129- echo reply [17], 133- router solicitation [18], 134- router advertisement [18] etc.
- Code: Its value depends on the message type. E.g. for message type= 1 (destination unreachable), **Table 1** shows the various code options.
- Checksum: 1's complement of sum of whole ICMP message is 1's complemented again to produce 16 bit checksum [17]. It is used to detect any message fault during transmission.

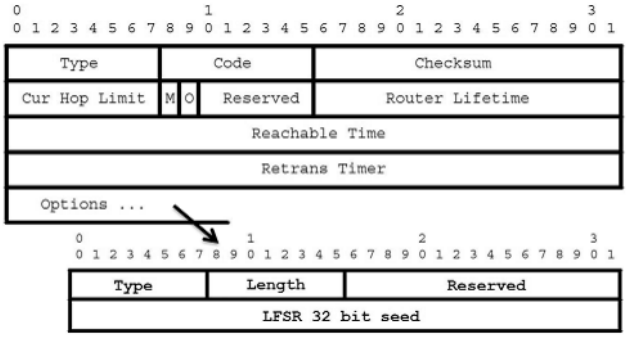


Figure 7. RA Frame Format

Table 1. Frame Code

Code	Name
0	no route to destination
1	communication with destination administratively prohibited
2	beyond scope of source address [17]
3	address unreachable
4	port unreachable
5	source address failed ingress/egress policy [17]
6	reject route to destination [17]
7	Error in Source Routing Header [19][20]

- Cur Hop Limit: Defines the maximum hop relay count for the message. Value= 0 means unspecified for this router and user is allowed to use any previous value.
- M: Managed address configuration flag. Value set means DHCP is available for address configuration. Also when set, makes 'O' flag redundant and can be ignored.
- O: Other configuration flag. Value set means other configuration parameters can be obtained from available DHCP server such as DNS information or some other server available on same network.
- Reserved: It is a 6 bit unused field for future use. Right now it should be filled with 0 (zero) and should be discarded at the receiver.
- Router Lifetime: Sending rules fix the upper limit value of this field to 9000 sec. It signifies the lifetime of router as default router. 0 (zero) in this field shows router is not default router and should not appear on the default router list.
- Reachable Time: This field contains value in milliseconds, tells a node about its neighbor reachability after receiving reachability confirmation.
- Retrans Timer: It signifies the time interval in milliseconds between consecutive neighbor solicitation messages. 0 (zero) value means it is unspecified by that router.
- Options: In general 3 types of options are available for RA. Each option contains separate fields for option type, length and value. Details of the three RA options are given in **Table 2**.
- In options we will add a new option for LFSR 32 bit seed. Its values would be:
  - Type= 6
  - Length= 1
  - Value= 32 bit LFSR seed

Table 2. RA Options

Type	Length (unit is 8 octets)	Option name
1	1	Source Link-Layer Address
3	4	Prefix Information
5	1	Maximum Transmission Unit

F. Pseudo code

- At RSU:
  - Input: initial seed, previous seed, bit1-bit4 location
  - Output: 32 bit seed

```
uint32_t rsu_seed (uint32_t &init, uint32_t &seed, uint32_t &bit1, uint32_t &bit2,
   uint32_t &bit3, uint32_t &bit4) //function to return 32 bit seed
{
   uint32_t bit;
   bit=((seed>>(32-bit1))OR(seed>>(32-bit2))    OR(seed>>(32-bit3))    OR(seed>>(32-
     bit4))) AND 1;
```

```
   seed= (seed>>1) OR (bit<<31);

   if(checkCycle(seed)==1) //cycle condition
   {
    clearSeed();
    init= time(NULL);
    seed= init;
     bit1= seed%10;
     bit2= (seed%100)/10;
     bit3= (seed%1000)/100;
     bit4= (seed%10000)/1000;
    return(rsu_seed(&init, &seed, &bit1, &bit2, &bit3, &bit4));
   }
   addSeed(seed);
   return(seed);
}
```

- Input: seed
- Output: ICMPv6OptionalHeader

```
IPv6_OPTION* seedHeader(uint32_t seed) //function for seed option header
{
   Ipv6OptionHeader *seedHdr;
   seedHdr->setType(ICMP_SEED_HEADER);
   seedHdr->setSeed(seed);
   returnseedHdr;
}
```

- At vehicle:
  - Input: RA
  - Output: IPv6 address

```
IPv6Address* receiveRA(IPv6_ICMP *packet)
{
   IPv6Address *ipv6_add;
   IPv6Header raHdr;
   IPv6OptionHeader prefixHdr, seedHdr;
   unit8_t type;
   uint64_thostID;
   packet->RemoveHeader(raHdr);

   while(packet)
   {
    packet->CopyData(&type, sizeof(type));
    switch(type)
    {
     case ICMP_SEED_HEADER:
      packet->RemoveHeader(seedHdr);
      hostID= (seedHdr.GetSeed())<<32;
      hostID= (hostIDOR MAC);
      ipv6_add->HostId(hostID, sizeof(hostID));
      break;

     case ICMP_PREFIX_HEADER:
      packet->RemoveHeader(prefixHdr);
      ipv6_add->Prefix(prefixHdr.GetPrefix(), prefixHdr.PrefixSize());
      break;

     default: //unknown option
      packet->RemoveHeader();
    }
   }
   return(ipv6_add);
}
```

The function rsu_seed() takes previous seed, bit positions and initial value as argument. In this function first we compute a new seed based on our LFSR algorithm. This seed is checked against the previously calculated seed values by checkCycle() function, if match found then initial value is again loaded with current system time and the previous stored seed is cleared by clearSeed() function. The new seed is added to the seed database and is returned for further processing. seedHeader() function takes the seed as its argument and create an Ipv6OptionHeader with this seed

value. This header is added to the original RA header and sent. receiveRA() function at vehicle side catches the RA and process it to extract the seed and network prefix. 64 bit hostID is computed by performing logical OR operation between MAC address and 32 bit left shift seed value. This hostID and extracted network prefix is copied to the hostID part and network prefix part of IPv6 address.

Performance Analysis

Simulation Setup

For analyzing the performance of the proposed mechanism, we have simulated the network architecture as explained in **Error! Reference source not found.** in ns3. Along with the LFSR based IPv6 mechanism, original MIPv6 (Mobile IPv6) has also been implemented .Comparison of Fibonacci LFSR with proposed LFSR and number of successful host ID configuration is also performed. The simulation parameters are as described in Table 3. Each simulation result is obtained by averaging fifty simulation rounds.

Results and Discussion

In Figure 8 we have compared basic Fibonacci LFSR with our modified LFSR for the number of seeds generated from both the algorithm. A seed round begins with loading initial value from system time and then generating the seed until either initial value or a previous generated value is encountered. For basic Fibonacci LFSR, result shows uniformity i.e. the output range around 1000 to 1050 rounds per seed. However for our modified LFSR the number of rounds per seed is non-uniform. The values are dynamic. For the basic Fibonacci LFSR, due to the constant bit position operation and uniform output a malicious node can guess the next seed which can be a potentially dangerous. But with the modified LFSR, since the bit position for operation changes with every round initialization and the number of rounds keeps varying which gives an additional security to the algorithm and protects the mechanism from guessing attacks.

Table 3 Simulation Parameters

Simulation area	5000m x 2000m
Simulation time	180 sec
Vehicle speed	5 km/hr – 100 km/hr
Communication range	1000 m
MAC	802.11 p
RAminDelay	200 ms
RAmaxDelay	1000 ms

Figure 9 shows the performance of our modified LFSR algorithm based on number of successful unique host ID configuration. For measuring this performance, single MAC ID is operated with different seed in different round to obtain a host ID. This step is repeated until new host ID clashes with already obtained host ID or the LFSR seed rounds up. While with the IEEE EUI host ID configuration we can configure only one host ID from a MAC ID which then needs to be checked for duplicate through DAD, with our LFSR based algorithm an average of 50 vehicles can be configured with different host ID for same MAC ID. With this result we emphasize on suppressing the need of DAD for our algorithm which reduces the time required in IP acquisition and ultimately in handoff latency.

Error! Reference source not found. show the performance of LFSR based IPv6 mechanism in terms of IP acquisition time with respect to vehicle speed and vehicle density. In **Error! Reference source not found.** (a), the vehicle is configured with different speed for each simulation run. The vehicle goes for handoff and the time needed to acquire IP is recorded for each configured speed. The IP acquisition time is calculated since the vehicle gets RA until it sends the HA registration message. The result also

includes the same performance graph for MIPv6 mechanism. The performance of MIPv6 gives a linear increment curve for increase in speed. Using our method, vehicle starting with a speed of 5 km/hr to 55 km/hr shows the same nature of linear increment but as the vehicle passes 55 km/hr speed the IP acquisition time shows a higher increase than the previous time. Further increase in speed increases the IP acquisition time by a bigger factor. This nature of output can be understood with the help of wireless channel properties i.e. as the speed increases, the wireless channel quality degrades. Bit Error Rate (BER) goes up and retransmissions are required which ultimately impacts the required time.

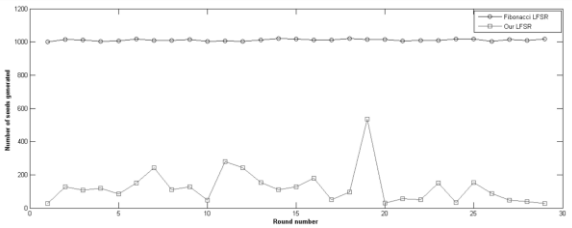


Figure 2. Fibonacci LFSR vs our LFSR

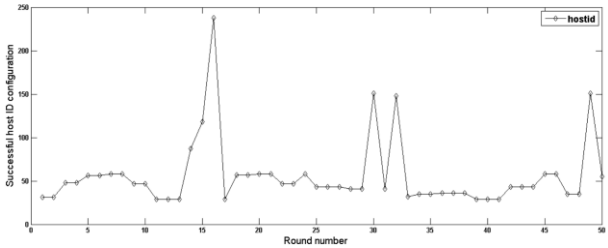
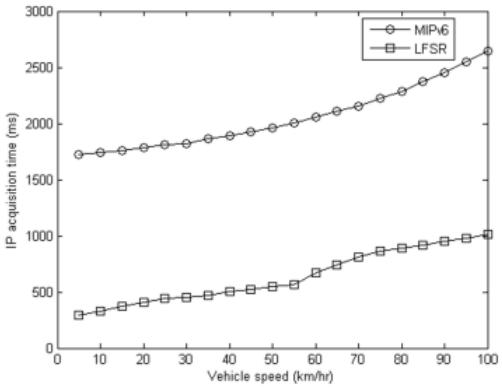


Figure 3. Successful hostID configuration with same MAC address

Error! Reference source not found. (b) shows the performance measured for IP acquisition time with respect to vehicle density. The more the density, the more will be time required for channel access.



(a)

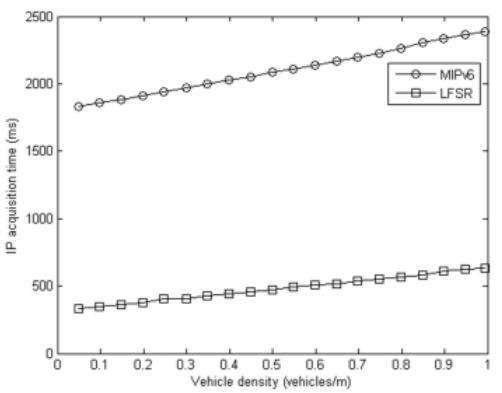
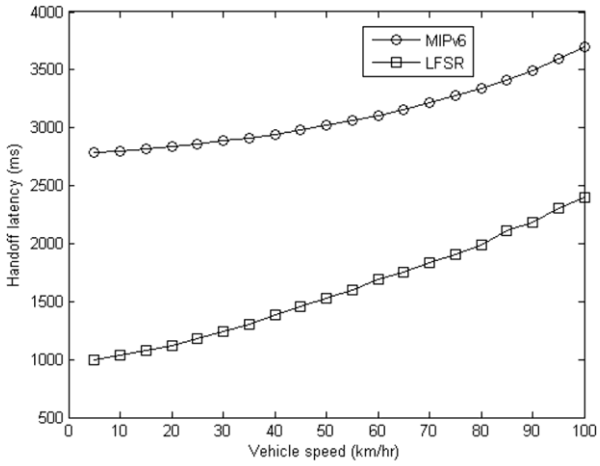


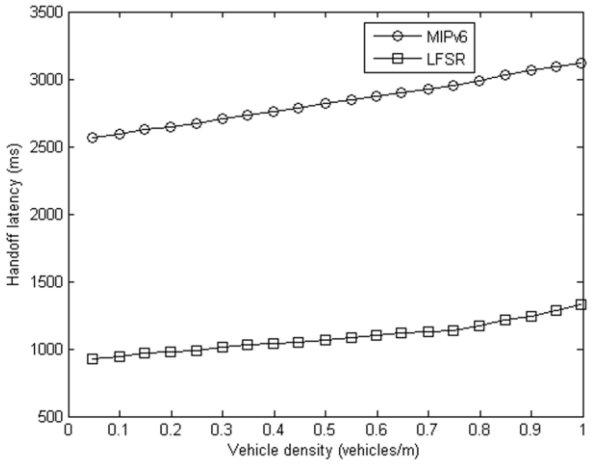
Figure 4 (a) IP acquisition time vs Vehicle speed (b) IP acquisition time vs Vehicle density

The result shows the same nature of linear increment with increase in density. Initially for 1 vehicle per 20 m, the time required is 335 ms which gradually increases to 344 ms, 365 ms and 378 ms for 1 vehicle/10 m, 1 vehicle/6 m and 1 vehicle/5 m. The same result nature is carried with further increase in vehicle density. Proposed algorithm outperforms MIPv6 for both vehicle speed and vehicle density as shown in the result.

While the previous result was concentrated to IP acquisition step only, Figure 5 shows the performance of whole handoff latency. The latency includes the time required by vehicle to execute all steps of handoff right from the triggering to HA registration. While Figure 5 (a) compares handoff latency with vehicle speed, Figure 5 (b) compares it with vehicle density. When comparing with vehicle speed, a single vehicle is run configured with different speed in each run. This vehicle goes for handoff in each simulation run and the time required for full handoff is recorded. A step of 5 km/hr speed is implemented in the simulation. For the initial speed of 5 km/hr a vehicle needs 1000 ms for a complete handoff. This latency increases with increase in speed like



(a)



(b)

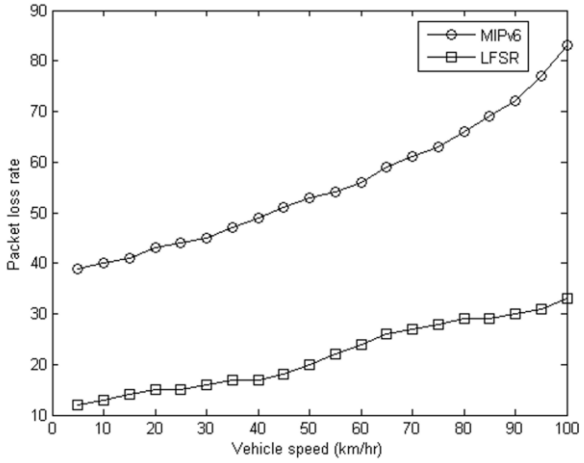
Figure 5. (a) Handoff latency vs Vehicle speed (b) Handoff Latency vs Vehicle Density

for 10 km/hr, 15 km/hr and 20 km/hr the resulting latency is 1042 ms, 1081 ms and 1122 ms. High latency for higher speed is due to nature of wireless channel. The channel fading, high BER and channel access delay are the major reasons for higher latency.

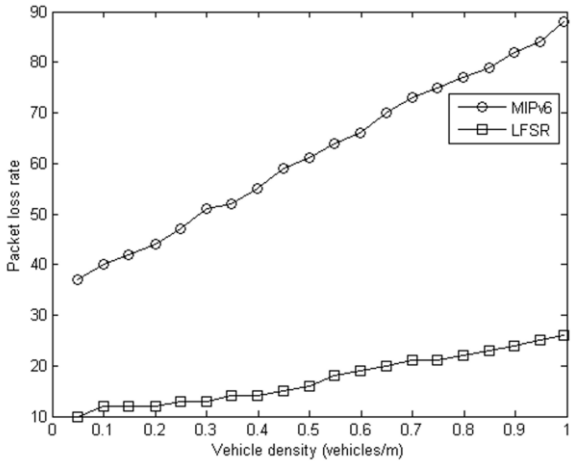
Handoff latency with respect to vehicle density doesn't show any significant change in handoff latency. For initial vehicle density of 1 vehicle/20 m, handoff latency is 927 ms. As the density increases further to 1 vehicle/10 m, 1 vehicle/6 m and 1 vehicle/5 m the latency increases to 941ms, 970ms and 977 ms. For higher vehicle density, handoff latency increases. Increase in vehicle density increases the channel access delay as well as interference which leads to retransmissions and adds additional delay to handoff latency. For the handoff latency also we have compared the proposed algorithm with MIPv6 and based on results, LFSR based IPv6 has outperformed MIPv6.

Result in Figure 12 (a) is calculated from packets missed divided by total transmitted packets with respect to vehicle speed and vehicle density. For initial speed of 5 km/hr, the packet loss rate resulted to be 12. The packet loss rate increases as the speed of vehicle increases. Once vehicle crosses the 45 km/hr speed, sudden increase in packet loss can be seen. Below 45 km/hr speed, the average packet loss rate increment was of 0.7 but after the speed is gained the average packet loss increment results to be of step 2. After 75 km/hr speed, the packet loss rate step decreases to 1. Due to increase in speed, various factors affect the performance of wireless channel and this leads to higher packet loss rate.

The next result in Figure 12 (b), packet loss rate is analyzed against the vehicle density. Result clearly shows that with increased vehicle density, the packet loss rate also increases. For initial vehicle density of 1 vehicle/20 m, packet loss rate stays around 10. Then as the density increases, proportional packet loss rate increment can be seen in the result. For 1 vehicle/10 m, packet loss rate is 11. Further the rate increases to 12, 12.5 and 13 for vehicle density of1 vehicle/6 m, 1 vehicle/5 m and 1 vehicle/4 m respectively. Major reason behind increased packet loss rate with respect to density is interference caused by other vehicles. However with respect to both vehicle speed and density, proposed algorithm performs significantly better than MIPv6.



(a)



(b)

Figure 6 (a) Packet loss rate vs Vehicle speed
(b) Packet loss rate vs Vehicle Density

Conclusion

In this paper, we have discussed a novel technique for IP acquisition in VANET. High mobility and vehicular density preclude the use of standard techniques. Due to unavailability of an optimal mechanism, VANET IP acquisition suffers from higher IP acquisition time which further increases the handoff latency. Situation is aggravated with link layer address conflict, which then needs manual intervention to configure. LFSR based IPv6 for VANET possess all the properties like lower IP configuration time and handoff latency, minimum packet loss etc. It even gives the solution for link layer address conflict situation thus removing the need of DAD and manual configuration. Performance of LFSR based IPv6 has been verified from simulation and packet traces. The research highlights are:

- It is favorable for high speed handoff.
- Works extremely well for higher density vehicle.
- Outperforms MIPv6 in all performance metric.
- Different unique host ID can be generated from same MAC, comes out to be an efficient alternative to EUI64 mechanism
- Provides novel mechanism for host ID configuration thus avoiding DAD and manual intervention.

- With feedback mechanism, additional security is introduced to avoid guessing attacks.
- Implementation does not needs any architectural changes in VANET but only modifies the control signal.

The above features makes LFSR based IPv6 for VANET suitable mechanism for IP acquisition. It is able to perform optimally in high vehicle density by avoiding link layer address conflict and thus DAD. With a significant support for high mobility and high density network, this mechanism can be adopted as a standard IP acquisition technique in VANET.

References

[1] R. Droms, H. Packard, T. L. B. Volz, C. Perkins, and M. Carney, “Dynamic host configuration protocol for ipv6 (dhcpv6)”, RFC-3315, July 2003.
[2] R. Droms, “Dynamic host configuration protocol”, RFC-2131, March 1997.
[3] M Fazio, M Villari, and A. Puliafito, "IP Address Autoconfiguration in Ad Hoc Networks: Design, implementation and Measurements", Computer Networks Journal, Elsevier Science Publisher, 50:898–920, 2006.
[4] S. Nesargi and R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network", In INFOCOM 2002, New York, June 2002.
[5] Y. Sun and E. M. Belding-Royer, "A Study of Dynamic Addressing Techniques in Mobile Ad hoc Networks", In Wireless Communications and Mobile Computing, pages 315–329, April 2004.
[6] S. Toner and D. Omahony, "Self-Organising Node Address Management in Ad-hoc Networks", In Personal Wireless Communications (PWC 2003), Venice, Italy, September 23-25 2003.
[7] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks", In ACM International Symposium on Mobile Ad Hoc Networking and Computing, Boston- Massachusetts, June 2002.
[8] K. Weniger, "PACMAN: Passive Autoconfiguration for Mobile Ad Hoc Networks", IEEE Journal On Selected Areas In Communications, 23(3), March 2005.
[9] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (aodv) routing", IETF Internet draft, MANET working group, June 2002.
[10] C. E. Perkins and E. M. Royer, "Ad hoc Networking, chapter Ad hoc On-Demand Distance Vector Routing", Addison-Wesley Publishers, 2000.
[11] D. B. Johnson, D. A. Maltz, Y-C. Hu, and J. C. Jetcheva, "The dynamic source routing protocol for mobile ah hoc networks (dsr)", IETF Internet draft, MANET working group, July 2004.
[12] M. Mohsin and R. Prakash, "Ip address assignment in mobile ad hoc networks", In Proceedings of IEEE MILCOM, September 2002.
[13] M. Fazio, C. E. Palazzi, S. Das, and M. Gerla, "Facilitating real-time applications in VANETs through fast address auto-con guration", in Proceedings of IEEE CCNC, pp. 981-985, 2007.
[14] Sadique Ahmed Bugti, Xia Chun He and Ejaz Hussain, "AutoCon guration for VANET, Integrated with Regional Code Association Architecture", 2012 4th International Conference on Computer Engineering and Technology (ICCET 2012) IPCSIT vol.40, IACSIT Press, Singapore, 2012.
[15] Kim H, Kim Y., "An early binding fast handover for high-speed mobile nodes on mipv6 over connectionless packet radio link", In Proceedings of Seventh ACIS International Conference on Software Engineering, Arti cial Intelligence, Networking, and Parallel/Distributed Computing, 2006.
[16] Hayoung OH, Joon Yoo, Chong-Kwon Kim and Sang Hyun Ahn, "VMIPv6: A Seamless and Robust Vehicular MIPv6 for Vehicular Wireless Networks and Vehicular Intelligent Transportation Systems (V-Winet/V-ITS)", Journal of Information Science and Engineering, vol. 26, pp. 833-850, 2010.
A. Conta, S. Deering and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
[17] T. Narten, E. Nordmark, W. Simpson and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
[18] P. Thubert Ed., A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012
[19] D. E. Culler, V. Manral, and J. W. Hui, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.
[20] Dimopoulou L, Leoleis G and Venieris IO, "Fast handover support in a wlan environment: challenges and perspectives", 19(3):14–20, IEEE Network 2005.
[21] Mark Goresky and Andrew M. Klapper, "Fibonacci and Galois Representations of Feedback-With-Carry Shift Registers", IEEE Transactions On Information Theory, Vol. 48, No. 11, November 2002.

