國立台灣師範大學資訊工程研究所碩士論文

幾何路由網路中有能源效率的
調變式 Beaconing 機制
An Adaptive Energy-efficient Beaconing Mechanism for Geographic Routing

研究生： Marko Durasic 杜馬克 撰

指導教授：蔡榮宗 博士

中華民國 103年 7 月

Abstract

In geographic routing, not much effort has been given to improving position update mechanism used by nodes in wireless Ad-Hoc networks which is an important issue in order to precisely maintain the geographic position of nodes. Nodes are periodically broadcasting beacon updates that contain geographic location coordinates to the nodes in their vicinity. This is an accepted scheme for maintaining neighbor's positions. We are solving the problem of high beacon overhead caused by generation of beacons while considering mobility and traffic load of the network. For example, if the beacon interval is too short for nodes that move very slow, i.e. not rapidly changing position, periodic beaconing will create unnecessary amount of beacon updates. In the same way, nodes which are broadcasting lot of data packets will generate the same amount of beacons as the nodes that have fewer data packet to transmit which leads to needless waste of energy. To make progress on these problems, we propose Adaptive Energy-Efficient (AEE) mechanism for geographic routing. Based on mobility and traffic load of the network, AEE adapts the beaconing interval to the value that corresponds to the network's demands. We have embedded AEE into well know Greedy Perimeter Stateless Routing protocol (GPSR), and compared it with regular GPSR. We ran extensive simulations in Network Simulator 2 (NS-2) for both networks with high and low initial energy level. Results have shown that AEE reduces beacon overhead up to 90 percent, AEE also reduces energy consumption and the number of packet collision up to 83 percent, without worsening packet delivery fraction of the network. This adaptive method that reduces the number of collision and prolongs the battery usage of each node in the network can be very useful for MANET or VANET where system is completely ad-hoc and no infrastructure is used and nodes are relying on their own power sources.

Keywords - Geographic routing, position update, beaconing mechanism, GPSR. Periodic beaconing, beaconing interval, MANET

Contents

Abstract i

List of Figures iii

List of Tables iv

1. Introduction 1

2. Related work 5

3. Adaptive Energy-efficient Beaconing Mechanism for Geographic Routing (AEE) 7

   3.1 AEE 7

4. Performance evaluation 14

   4.1 Results showing the impact of AEE on various node speed and various numbers of data flows with high initial energy 17

   4.2 Results showing the impact of AEE on various node speed and various numbers of data flows with low initial energy 28

5. Conclusion 43

References 44

List of Figures

Figure 1*:* Local Minima Problem (left) and Node x's void with respect to destination D. (right)[*4]* 2
Figure 2*:* The right-hand rule (interior of the triangle). x receives a packet from y, and forwards it to its first neighbor counterclockwise about itself.[*4]* 2
Figure 3. Simulation results showing effect of network mobility on Packet Delivery Fraction 18
Figure 4. Simulation results showing effect of number of data flows on Packet Delivery fraction 18
Figure 5. Simulation results showing effect of network mobility on Average Hop Count...........................20
Figure 6. Simulation results showing effect of number of data flows on Average Hop Count 20
Figure 7. Simulation results showing effect of network mobility on Average End to end Delay...............21
Figure 8. Simulation results showing effect of number of data flows on Average End to end Delay 22
Figure 9. Simulation results showing effect of network mobility on Beacon Overhead 23
Figure 10. Simulation results showing effect of number of data flows on Beacon Overhead 23
Figure 11. Simulation results showing effect of network mobility on number of packet collisions 24
Figure 12 Simulation results showing effect of number of data flows on number of packet collisions.......25
Figure 13. Simulation results showing effect of network mobility on Energy Consumption 26
Figure 14. Simulation results showing effect of Number of Data Flows on Energy Consumption 26
Figure 15. . Simulation results showing effect of network mobility on number of received data packets......29
Figure 16. Simulation results showing effect of number of data flows on number of received data packets 30
Figure 17. Simulation results showing effect of network mobility on Packet Delivery Fraction.................31
Figure 18. Simulation results showing effect of number of data flows on Packet Delivery Fraction 31
Figure 19. Simulation results showing effect of network mobility on Average Hop Count.........................32
Figure 20. Simulation results showing effect of number of data flows on Average Hop Count 33
Figure 21. Simulation results showing effect of network mobility on Average End to end Delay...............34
Figure 22. Simulation results showing effect of number of data flows on Average End to end Delay 34
Figure 23. Simulation results showing effect of network mobility on Beacon Overhead 35
Figure 24. Simulation results showing effect of number of data flows on Beacon Overhead 36
Figure 25. Simulation results showing effect of network mobility on number of packet collisions 37
Figure 26. Simulation results showing effect of number of data flows on number of packet collisions......37
Figure 27. Simulation results showing effect of network mobility on Energy Consumption 38
Figure 28. Simulation results showing effect of Number of Data Flows on Energy Consumption 39
Figure 29. Simulation results showing effect of network mobility on number of shut nodes......................40
Figure 30. Simulation results showing effect of number of data flows on number of shut nodes 40
Figure 31. Simulation results showing effect of network mobility on number of shut nodes throughout the simulation time...41
Figure 32. Simulation results showing effect of number of data flows on number of shut nodes throughout the simulation time...42

List of Tables

Table 1. Variables used in the algorithms 13
Table 2. Parameters of the first set of simulations 17
Table 3. Parameters of the second set of simulations 28

1. Introduction

Geographic routing protocols for ad-hoc networks are becoming more and more popular within the research community. One of the reasons for their growing popularity is the fact that in regions in which there is not much or no communication infrastructure, or the existing infrastructure is expensive and not convenient to use, wireless nodes may still be able to communicate and send packets because of the nature of ad-hoc networks. This kind of system is suitable for VANET [1] and MANET.

Wireless Ad- Hoc networks do not require communication infrastructure. A wireless ad hoc network is a decentralized type of wireless network [2]. *Ad hoc* is a Latin word and it means "for this purpose" [3]. Routers are needed in wired networks and access points are needed in infrastructural wireless networks, but not in a Wireless Ad- Hoc network. Each node contributes in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically using some kind of routing protocol.

One kind of Wireless Ad- Hoc network is mobile ad hoc network (MANET). Nodes in MANET can move independently of each other, but all of the nodes can be used in forwarding data. While moving, nodes create new links with the nodes that are within their transmission range frequently.

Geographic routing is a kind of routing principle that relies on geographic position information. Main idea of geographic routing is that nodes are using geographical position to choose next hop node. In geographic routing we assume that nodes are aware of their own position. Position can be obtained with some sort of position navigation system (e.g. GPS). Before making forwarding decision, geographic location of the destination node and neighboring nodes is necessary. The destination of a packet is the final node to which the packet is traversing. Neighboring nodes are the nodes that are within the limits of radio transmission range of the node that is making the routing decision. Local topology is a local chart that each node creates and it consists of nodes that are in the nodes vicinity.

One of the most popular geographic routing protocols is Greedy perimeter stateless routing (GPSR) [4].In GPSR the routing decision is made using merely local topology, which means there is no global topology. This kind of characteristics makes geographic routing completely stateless. Each node keeps track of its neighboring nodes by maintaining a neighbor table [4]. Greedy forwarding of data represents the method of deciding to which node to transmit data. GPSR operates in greedy forwarding manner where a node transmits data to a neighboring node that is geographically closest to the destination. However GPSR can be stuck in local minima situation (Figure 1), when there is no any neighboring node closer to the destination than the node that has data packets to transmit. In this situation GPSR recovers by routing around the perimeter of the void where greedy forwarding is impossible (Figure 1).

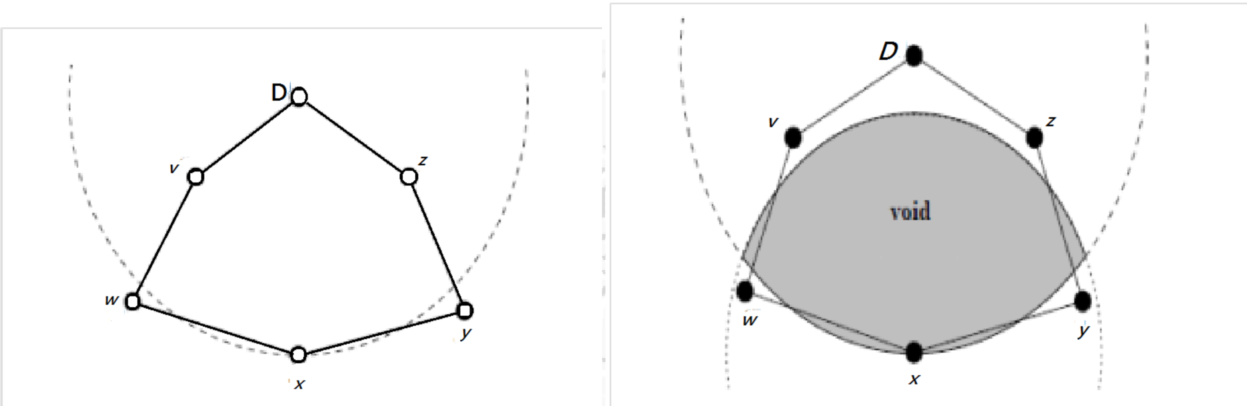

Figure 1*: Local Minima Problem (left) and Node x's void with respect to destination D. (right)[*4*]*

It uses the right hand rule for traversing a graph which is depicted in Figure 2.

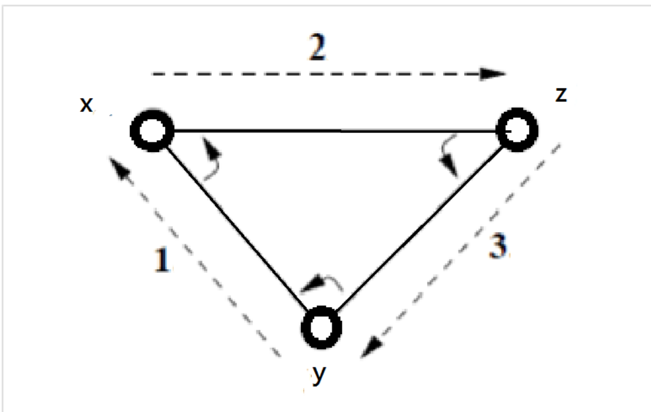

Figure 2*: The right-hand rule (interior of the triangle). x receives a packet from y, and forwards it to its first neighbor counterclockwise about itself.[*4*]*

Local topology needs to be maintained with frequent beacons updates generated by nodes in the network. Beacon packets contain node's position information and its identifier. Nodes broadcast beacon packets. This can be done periodically or the mechanism can be made entirely reactive. Periodic beaconing is a simple beaconing algorithm where periodically, each node transmits a beacon message. Reactive beaconing is a sort of on demand beaconing where nodes ask for beacons with a broadcast "neighbor request" only when they have data packets to send.

Periodic beaconing yields numerous problems and it can generate a large number of beacon messages which leads to high beacon overhead. Every node in the network sends position updates even to the nodes that are still or not transmitting data. This can worsen routing performance and produce numerous beacon packets, resulting in high update cost in high-mobility networks. Also, nodes that are highly mobile will not maintain accurate network topology due to the rapid changing of network topology.

Reactive beaconing procedure starts before each data transmission [5]; as a result, for networks that have a lot of source destination pairs, many unnecessary beacon messages are broadcasted. Furthermore, increased numbers of beacons lead to higher energy consumption at each node as nodes consume additional energy for transmission and reception of packets; as well it increases the probability of collision at the medium access control (MAC) layer. Packet delivery ratio, overall routing performance, beacon overhead and end-to-end delay are worsening by these problems. Having all these mentioned above it is useful to adapt the beaconing mechanism somehow.

We propose an adaptive beaconing mechanism called *An Adaptive Energy-efficient Beaconing Mechanism for Geographic Routing (AEE)*. In further reading, when referring to our beaconing mechanism we will be using the abbreviation *AEE*. Our beaconing mechanism adapts the beacon interval to the value ΔT based on the function of mobility of nodes and the number of data flows that are passing through that node. Nodes that are not extremely popular *i*.e., nodes that are not frequently transmitting data packets and the nodes that are not very mobile will have long beacon interval. This makes higher mobile and popular nodes generate more frequent beacon updates than the nodes that are mostly static and non-popular. This lowers the beacon overhead and energy consumption.

We run a series of simulation to evaluate our strategy and to determine what sort of impact it makes on packet delivery fraction, beacon overhead, end to end delay, energy consumption, number of packet collisions in MAC layer, hop count and other network metrics.

The rest of the thesis is organized as follows: In Section 2, we discuss some related work. A thorough explanation of the AEE mechanism is given in Section 3. In Section 4 we present the results of our simulations showing performance improvements succeeded by a conclusion of our thesis in Section 5.

.

2. Related work

Mass majority of proposed research in the field of geographic routing is dealing with the strategies that are used to locate the destination of the packets that are being routed, but not a lot of effort was given in researching the improved approach to more accurately locate neighbor's geographic position. Nodes update their neighbor tables with needed information about neighboring nodes. In Self-Adaptive On-Demand Geographic Routing Protocols for Mobile Ad Hoc Networks [6], topology-based and geographic combined mechanism to reactively search for next hop is proposed. In their scheme nodes would request messages reactively (*i*.e., only when they have data to transmit) and wait for a certain backoff interval. In case that there are no neighbors that are closer to the destination, the local minima problem will appear. Instead of switching to perimeter routing as in GPSR[4] , a different recovery strategy with expanded ring search to two hop neighboring environment has been introduced in [6] [7]. When a 2-hop neighbor receives a beacon message and its closer to the destination than the nodes that generated the beacon, it will send a reply following the same path in reverse mode. In their other scheme (SOGR with Geographic Reactive Mechanism) the periodic beaconing is only triggered when a node overhears data traffic, and the beaconing will stop if no traffic is heard for a pre-defined period. This approach however does not the take the number of data flows into consideration, and that is crucial if the reactive beaconing strategy is applied. Networks with a lot of source destination pairs would produce unnecessary large numbers of beacon which would then lead to a high beacon overhead.

In Adaptive Position Update for Geographic Routing in Mobile Ad-hoc Networks [8] the beacon intervals are being adapted based on mobility of nodes and traffic patterns in the network using Mobility Prediction Rule (MP), so highly mobile nodes would generate more beacons. As they assume that all one-hop neighbors operate in the promiscuous mode, they can overhear the data packets that are being transmitted in their vicinity. A node broadcast beacons when it overhears data transmission. This On Demand Learning Rule maintains more frequent beacon updates along the routing paths, but in networks with many data flows it still creates a lot of needless beacon messages. As well, nodes that are far away from the data flow path would still send beacon updates. In fact, it would be more suitable for those nodes to adapt their beacon intervals based on their mobility and data flows to consume less energy and decrease collision probability.

As an alternative, we propose a beaconing mechanism that can reduce beacon overhead and adapt more suitably to wireless networks with high mobility and many data flows.

3. Adaptive Energy-efficient Beaconing Mechanism for Geographic Routing (AEE)

In geographic routing protocols such as GPSR, the nodes are sending and receiving beacons that contain node position periodically and then updating their neighbor table. Based on that neighboring table the selection of next hop node is made. In our research we assume that each node knows its own speed and position.

Based on the speed and position, node can adapt its beaconing period to ΔT.

3.1 AEE

Highly mobile nodes are more likely to leave the transmission range of their neighboring nodes; hence the beacon interval should be adapted to allow the neighboring table more precisely updated for this kind of nodes. Our research has shown that highly mobile nodes should have shorter beacon intervals. Additionally, nodes that are not moving or moving slowly should not have a short beacon interval since it is unlikely for them to move out of their neighbor's transmission range and therefore their position can remain accurate for a long amount of time. Moreover those static nodes can adapt their beacon periods following the demands of the network thus saving a lot of battery power. Without reducing beacon overhead a large number of data flows passing through specific nodes can lead to increased possibility of collisions and result in inaccurate location information.

Therefore, we introduce an adaptive mechanism where beacon interval ΔT is adapted frequently. If nodes send beacons periodically and there are a lot of nodes in the network, this method would seem to create a lot of beacon overhead for regular non-adaptive beaconing mechanism. However our adaptive beaconing mechanism solves this problem.

Nodes can keep track of the number of data flows that traverse through them by incrementing the value ***df*** by 1 every time the node is found on a new data path. That information is held in each node's data stack. Here we are using a method for organizing and controlling data stack, where the oldest (first) entry is removed first when the number of entries in the stack becomes greater than the MAX value. The method is explained in the following pseudo code.

```
1    function Initialize
2      stack_pointer = 0
3        df =0
4        set time_limit
5        Let M[0..MAX] and T[0…MAX] be new stacks
6        for i=0 to MAX
7          M[i] = 0
8          T[i] = 0
9        end for
10       return stack_pointer, df, time_limit, M, T
11     end function
```

Function **Initialize** is called at the initialization of the network. This procedure just sets the value of the *stack_pointer and df* to zero in lines 2-3, as we are assuming that the array elements of the stack begins at 0 and no traffic has yet flowed through the node. Variable *time_limit* is also created in line 4. This variable will be used to remove outdated entries from the stack. It is also creating two stacks of size MAX and it fills them with zeros in lines 5-9. One is used for tracking source destination pairs, another one for storing source destination pair’s time-out intervals. We are also assuming that the maximum number of elements of the stack is MAX. Below we can see how these variables are used.

```
1  function maintain _df (M, newpacket,df)
2    for j =0 to MAX
3       if(newpacket    != M [j])
4                Boolean value = TRUE
5            Else
6                Boolean value = FALSE
7                          Break
8       end if
9    end for
10    If (stack_pointer <MAX)
11            If (Boolean value = TRUE)
12                      M[stack_pointer] = newpacket
13                      T[stack_poiinter]= time_limit # decrement by one each sec
14                      stack_pointer = stack_pointer + 1
15                        If ( df< MAX)
16                             df = df +1
17                      else
18                           do nothing
19            else
20                      break
21            end If
22    else
23               stack_pointer = stack_pointer - MAX;
24                  M[stack_pointer] = new packet
25                  stack_pointer = stack_pointer +1;
26    end if
27  return df
28 end function
```

In detail, the functions ***maintain _df*** works as follows: The for loop of lines 2-9 checks for previous entries in the stack and sets the *Boolean value* to true or false depending on the presence of the packet with same source destination information as the previous entries in the stack. If the packet is coming from a different data flow, then the *Boolean value* is set to true, otherwise it is set to false in which case break command is invoked and the program gets out of the loop. Line 10 checks if the stack_pointer is larger than the maximum size of the stack MAX. If it’s not that means that there are still empty slots to write new SD pairs. Line 11 checks if the Boolean value is true or not. TRUE value represents that the new coming packet is from a different SD pair; therefore lines

12-18 are executed. Line 12 writes the new SD pair to the stack at the position where the stack pointer is “pointing”, i.e. available empty slot. Line 13 starts the timer for that specific entry. After that the stack pointer is incremented by one in line 14. Stack pointer is incremented by one so it can point to the next empty slot; df is also incremented by 1 because the new data flow has been written to the stack; *df is* incremented only if the current value of df is smaller than MAX, as MAX represents the maximum size of the stack therefore it is impossible to store more information about number of data flows. This way we can keep track of the number of most recent data flows passing through each node. Line 20 is executed in case the Boolean value is set to *FALSE* in line 6, which means that the new coming packet is a member of an SD pair that has been written to the stack already, so we can call command *break* in line 20 to get out of the if statement as there is no need for inputting anything to the stack. If the value of the stack_pointer is higher than the MAX size of the stack (which means stack is full), stack_pointer value is decremented by MAX in line 23. So the stack_pointer will point to the first entry of the stack and we can write to that position. By decrementing value of the stack pointer we are making sure that the pointer is pointing to the oldest entry of the stack so that the new value of SD pair can be written in line 24. Using this technique we are providing the stack with the most relevant SD pairs. In line 25 stack_pointer is incremented by 1 so it can point to the next oldest entry of the stack. Line 26 returns variable *df* so it can be passed on to the next function called ***time_out.***

After inputting new source destination pairs to the stack we are calling function *time_out* to check the stack for the expired entries, i.e. entries that have been in the stack longer than the pre-set *time_limit:*

```
1   function time_out (M,T)
2     for i =0 to MAX
3       if ( T[i]=0)
4           M[i]=0
5          df= df-1
6       else
7           do nothing
8       end if
9     end for
10  return df
11  end function
```

Entries are being checked in lines 2-9. Nodes whose timer has reached zero will be removed, i.e. set to zero. After removing outdated information in line 4, df is decremented by one to reflect the changed number of relevant data flows in line 5. Function *time_out* returns df in line 10 so it can be passed to the next function: ***deltaT***.

Nodes can calculate their ∆T based on the algorithm below. We are assuming that the nodes know their own speed.

```
1   function delta_T (df, s)
2     ΔT = (1 / ((1+df) * (1+s))) *C
3           if (ΔT < min)
4               ΔT = min
5     return ΔT
6   end function
```

The formula in line 2 calculates beaconing period. ∆T is a function of number of Source-Destination (SD) pairs and node speed; df denotes the number of SD pairs, s denotes speed and C denotes a constant that we use to adapt the maximum value of ***ΔT***. This means that a higher number of SD pairs will result in a shorter beacon period, and vice versa. Higher speed will result in shorter beacon period. On the other hand, very low values of ΔT or value 0 would badly affect the network performance for various reasons such as increased collisions, beacon overhead, etc. For that reason the minimum value is set in lines 3-4.

Functions **Initialize, *time_out, maintain_df*** and ***deltaT*** are called at the initialization of the network. After initialization, functions ***maintain_df***, ***time_out*** and ***deltaT*** are called at the arrival of the new packet. Functions ***time_out*** and ***deltaT*** are also called every ΔT period of time, in case there are no changes in data traffic activity but there are chances in mobility. All of the used symbols are listed and defined in Table 1.

It is important to adequately control the maximum value of the adapted beacon interval as excessively high value of ΔT will not contribute to improved performances of the routing mechanism. After initializations and calculation of ΔT, nodes broadcast beacons periodically, and each node's beacon interval is set to ΔT respectively, based on the speed and number of data flows traversing through it. Depending on the mentioned factors, beaconing interval will be higher or lower. If a node is "unpopular" or "safe", i.e. it is moving slow, or it is not found on the data path frequently, its value of ΔT is set to a long interval. After the expiration of this beaconing period (ΔT), ΔT is calculated again using the current speed and data flow information. This means that after broadcasting beacons for some time, in the case of speeding up or sending lot of data packets frequently, the node's value of ΔT will become lower and it will become a "popular" and "unsafe" node. This means that its periodic beaconing interval ΔT is shorter, therefore broadcasting beacons more frequently.

This kind of beaconing strategy ensures that beacon intervals for static and "unpopular" nodes stay longer thus generating less beacon packets but still sustaining an accurate local topology. On the other hand those nodes which are extremely mobile and frequently transmitting data will frequently broadcast beacon packets as well.

Symbols	Definition
Initialize	Function called at the initialization of the network
stack_pointer	Pointer variable that "points" to an empty slot in the stack
df	Number of data flows
time_limit	Life time of each entry in the stack
M	Array that holds the info about source destination pairs. First column holds the source information and second one holds the destination information
T	Array that holds the information about the period of time each node has spent in the stack
maintain_df	Functions used to maintain the stack and track new data flows
newpacket	Array that holds the source and destination of the new coming data packet
Boolean Value	Boolean value which represents the presence of the new source destination pair in the existing table
MAX	Maximum size of the stack
time_out	function used to remove outdated entries from the stack
delta_T	function used to determine adapted beacon interval
s	Speed of the node
ΔT	Adapted beacon interval
C	Constant number
min	Minimum value of ΔT

Table 1. Variables used in the algorithms

4. Performance evaluation

In this section we are presenting a set of simulations that we have done in a popular network simulator tool called NS-2[9]. We used the following metrics for our evaluations: Packet delivery fraction, average end to end delay, total beacon overhead, total energy consumption, average hop-count, packet collisions**,** number of received packets and number of shut nodes.

Each network metric is explained as follows:

1. **Packet Delivery Fraction**: the fraction of the number of delivered data packet to the destination. This shows the proportion of delivered packets to the destination.

   Packet Delivery Fraction = ∑ Number of packet received / ∑ Number of packet sent.

   The greater value of packet delivery fraction means the better performance of the routing protocol.

2. **Average End-to-end Delay:** the average time that takes a data packet to arrive at the destination. This is caused due to the delay of route discovery process and the line in data packet transmission. Only the data packets that are successfully delivered to destinations are counted.

   ∑ (arrive time – send time) / ∑ Number of data packets received by the destination nodes

   The lower value of end to end delay means the better performance of the protocol.

3. **Total Beacon Overhead**: the total number of beacon packets transmitted by all nodes in the network. The lower value of total beacon overhead means the better performance of the network, provided that Packet Delivery Fraction and other important metrics are not negatively affected by it.

4. **Total energy Consumption**: the sum of energy consumed by each node in the network. Energy is consumed during transmission and reception of packets. The lower value of total energy consumption means the better performance of the network, provided that Packet Delivery Fraction and other important metric are not negatively affected by it.

5. **Average hop-count**: The **hop count** refers to the number of nodes through which data must pass between source and destination. Each node along the data path represents a hop. The hop count is therefore a basic measurement of distance in a network.

6. **Packet collisions:** the number of packet collisions at the Medium Access Control (MAC) layer.

7. **Number of received packets:** the number of delivered data packet to the destination.

8. **Number of shut nodes:** the number of nodes that have lost all the available energy and are incapable of transmitting or receiving data

Results are showing impact of our Adaptive Energy-Efficient Beaconing Mechanism (AEE) for Geographic Routing on the above mentioned metrics. We tested our mechanism on GPSR [4], and compared results of GPSR with AEE (GPSR-AEE) and a regular GPSR. We will be referring to GPSR with AEE as GPSR-AEE. The simulations were performed for simulation time of 900 seconds. In every simulation run, 50 nodes were positioned randomly on the field of size 1500m×600m. Each node transmission range is 250m. Data packet size is 64 bytes and each source is using CBR (Constant Bit Rate) that generates 4 packets per second. In both of two sets of simulations we have chosen 15 random source-destination pairs as data-flows and we varied the speed from 1m/s to 30m/s, so we can estimate the impact of mobility on network performances. We also varied the number of data flows from 1 to 30, thus showing the effect of different number of data-flows on the network performances. In the first set of simulations we have tested our mechanism for high values of initial energy that every node has. In the second set of simulations we have tested our mechanism for low values of initial energy that every node has. In the former, the energy level of each node has been set to be enormously high at 1000 Joules at which the node will supposedly never run out of energy. In the later, it has been set very low at 10 Joules, so we can observe the influence of our energy-efficient algorithm on nodes shutting down due to the energy loss.

4.1 Results showing the impact of AEE on various node speed and various numbers of data flows with high initial energy

Values and parameters which are used in this simulation are presented in Table 2:

Parameters	Value
Number of nodes	50
Topology	1500 m X 600 m
Duration of the simulation	900s
Transmission range	250 m
Maximum speed of the nodes	0~30 m/s (15m/s in simulation where we tested AEE for various numbers of data flows)
Number of Data Flows	1~30 (15 data flows in simulations where we tested AEE for various node speed)
Mobility model	Random Waypoint Model
Traffic Source	Constant Bit Rate (4 packets/second)
Data Packet Size	64 bytes
Initial energy level of each node	1000 Joules

Table 2. Parameters of the first set of simulations

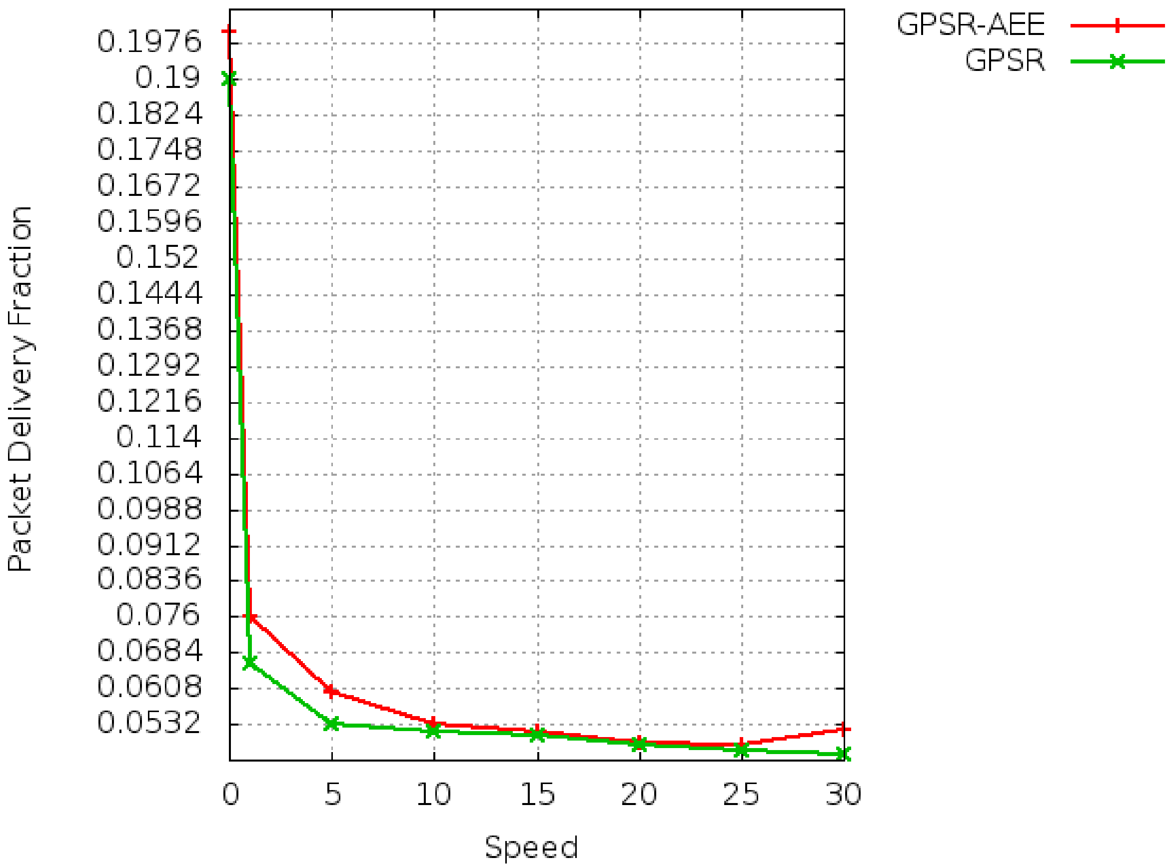

Figure 3. Simulation results showing effect of network mobility on Packet Delivery Fraction

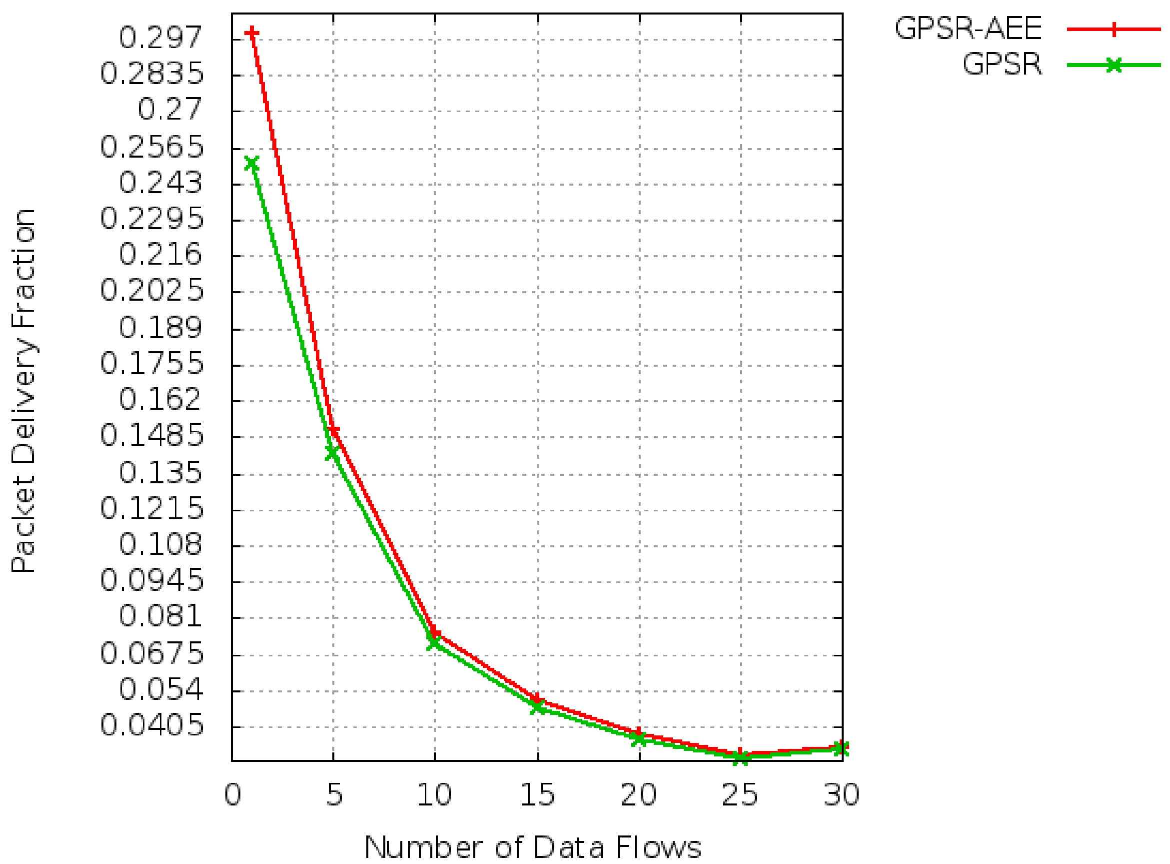

Figure 4. Simulation results showing effect of number of data flows on Packet Delivery fraction

Figure 3 and Figure 4 show measurement of packet delivery fraction for two scenarios of various speeds and various numbers of data flows respectively. Packet delivery fraction of GPSR-AEE is showing improvements comparing to GPSR in both scenarios. As our mechanism is of adaptive nature more accurate network topology is maintained.

Figure 3 illustrates that for low mobility GPSR-AEEE achieves higher values of the packet delivery fraction. However, at higher speeds GPSR-AEE and GPSR show similar values of the packet delivery fraction. Higher speed of nodes makes them move away from their neighbors' transmission range more frequently, thus it is harder for nodes to maintain accurate topology of their neighbors.

Figure 4 illustrates that for smaller number of data flows GPSR-AEE slightly outperforms GPSR. This improvement is caused by more accurate network topology that is maintained by GPSR-AEE beaconing mechanism. More accurate topology means more alternate routes available. For higher number of data flows both GPSR-AEE and GPSR show similar results. This performance result is attributed to the fact that data packets piggyback the local sending node's location on all data packets it forwards, and runs all nodes' network interfaces in promiscuous mode, so that each node gets a copy of all packets for all nodes within radio range [4].Therefore, increased traffic load leads to more accurate topology of the network as location updates are transmitted more frequently..

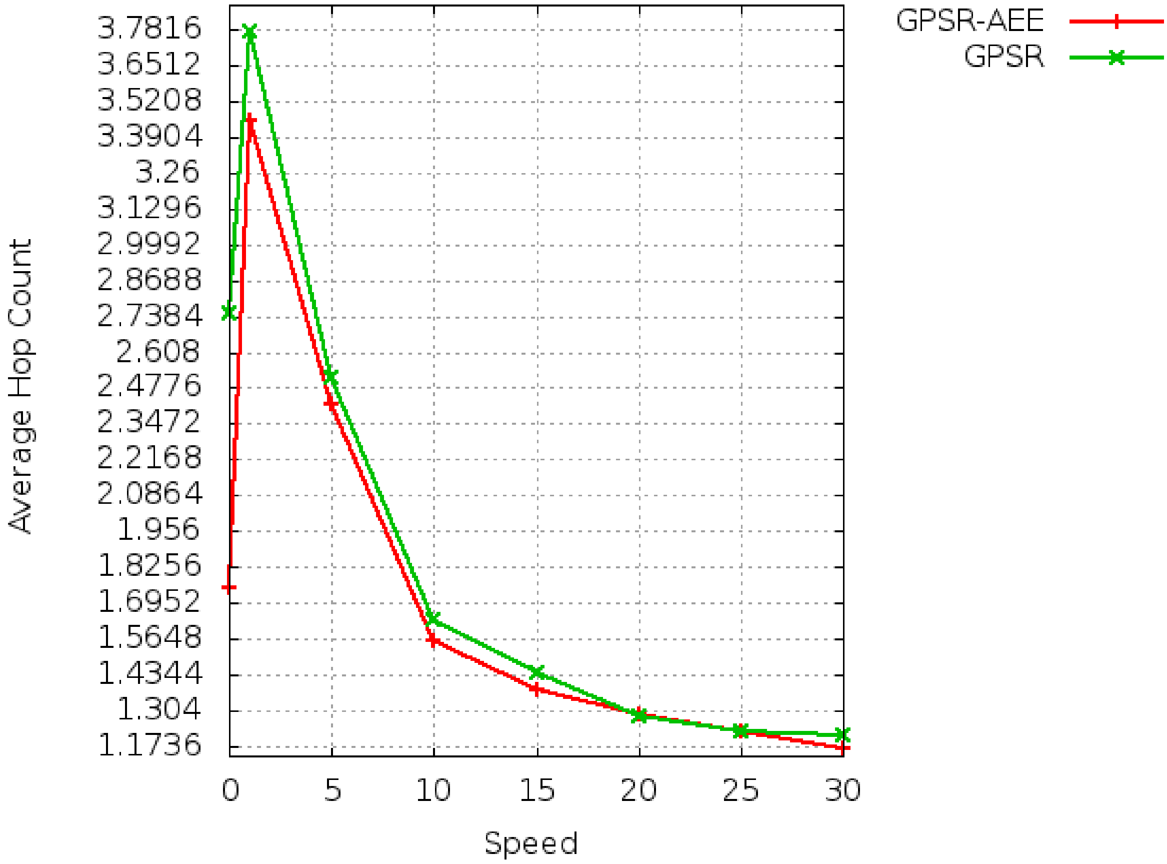

Figure 5. Simulation results showing effect of network mobility on Average Hop Count

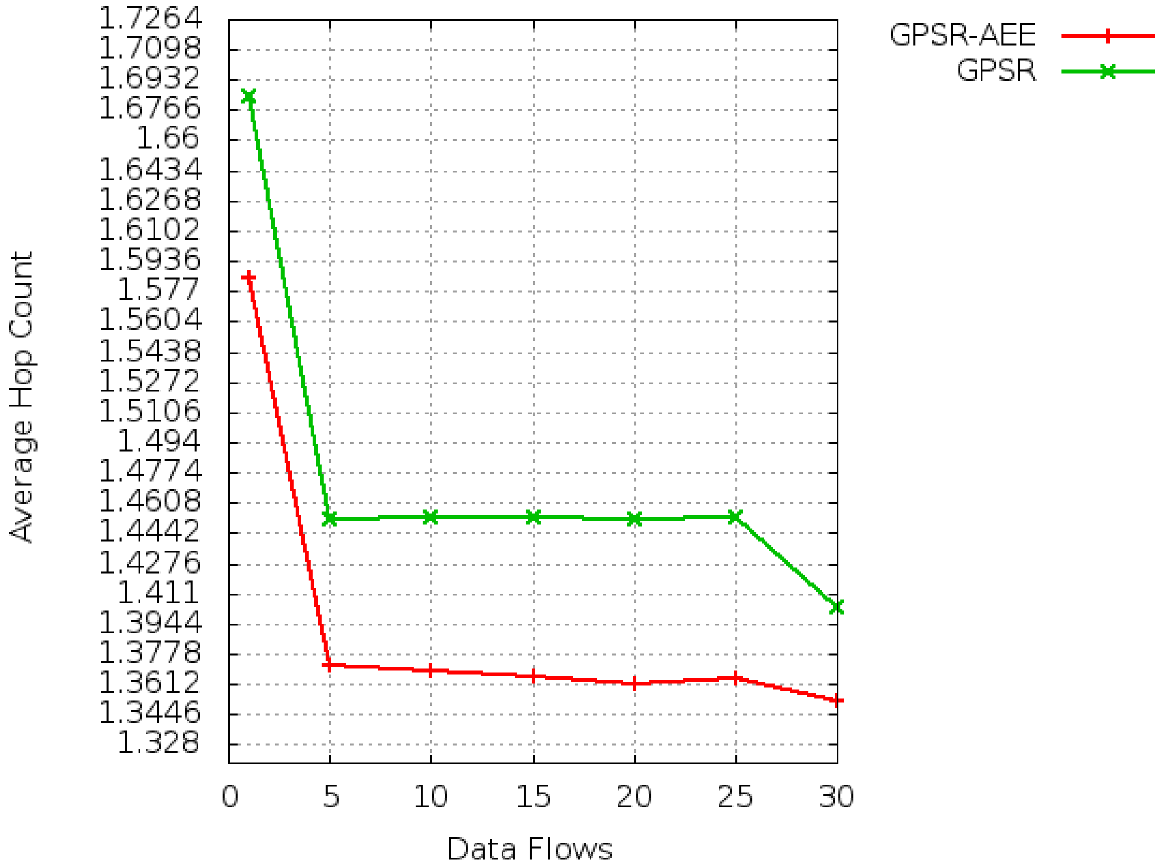

Figure 6. Simulation results showing effect of number of data flows on Average Hop Count

By maintaining a more accurate network topology, GPSR-AEE is able to route packets through fewer nodes than GPSR. Figure 5 illustrates the effect of network mobility on Average Hop Count. The reduced hop-count at low mobility is mainly due to the MAC-layer collisions illustrated in Figure 11. For higher mobility GPSR-AEE reduces the hop-count due to the fact that periodic beaconing used in GPSR is not sufficient in maintaining an accurate network topology. For example, node can often send a packet to its neighbor, which is not anymore in its transmission radius.

Figure 6 depicts the effect of varying the number of data flows on Average Hop Count. For low mobility hop-count is reduced due to the lower number of MAC-layer packet collisions illustrated in Figure 12. For higher mobility, the additional number of beacons generated by GPSR-AEE is used to maintain more accurate topology. This provides more precise route for packets to traverse to the destination.

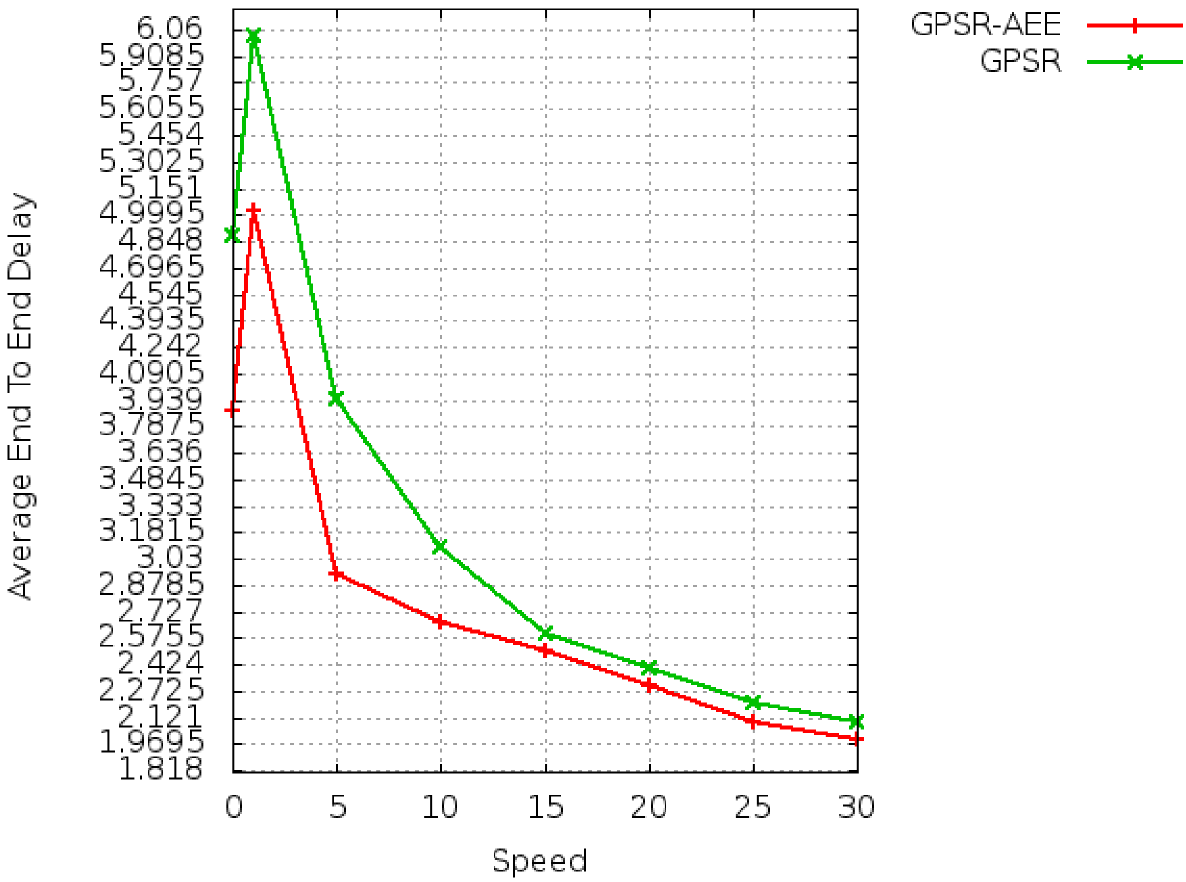

Figure 7. Simulation results showing effect of network mobility on Average End to end Delay

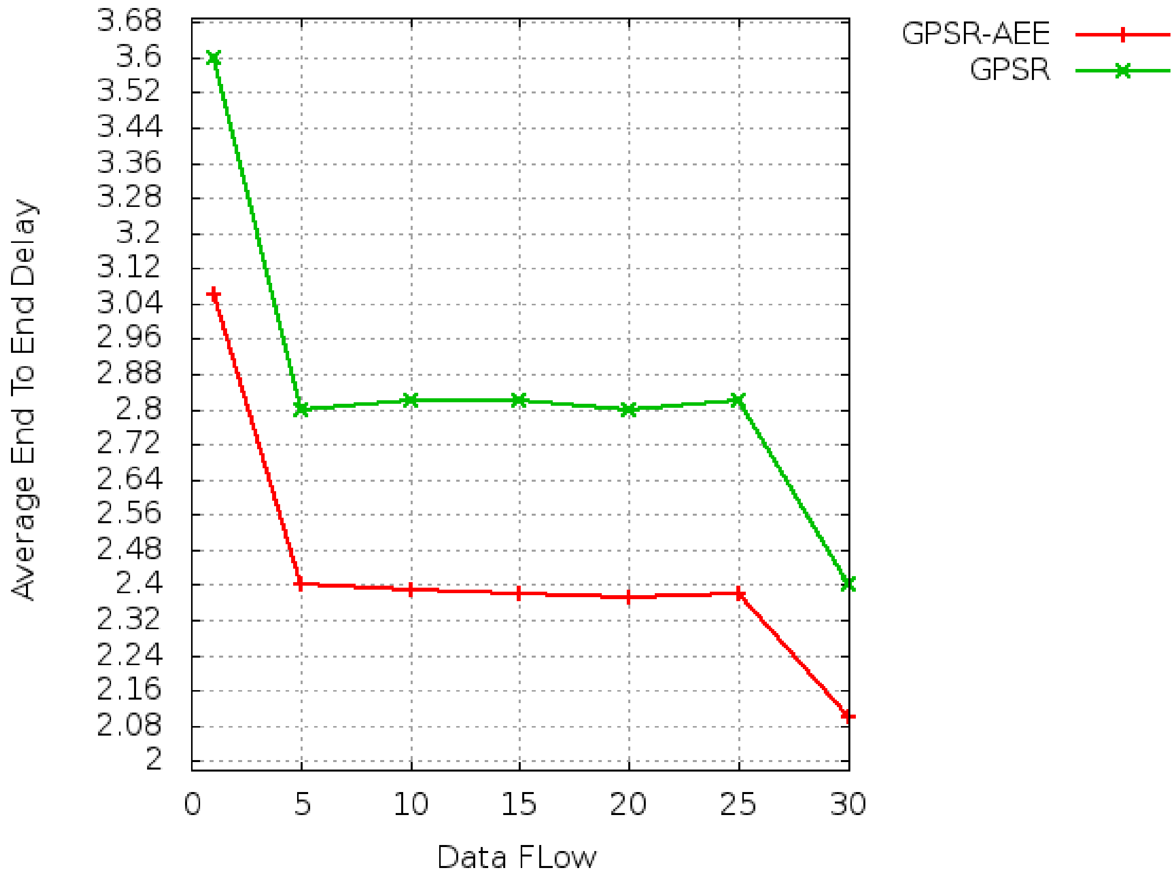

Figure 8. Simulation results showing effect of number of data flows on Average End to end Delay

In figures 7 and 8 we can see that GPSR-AEE can reduce average end to end delay too. These results are proportional to average hop-count results. A higher average hop-count produces higher end to end delay.

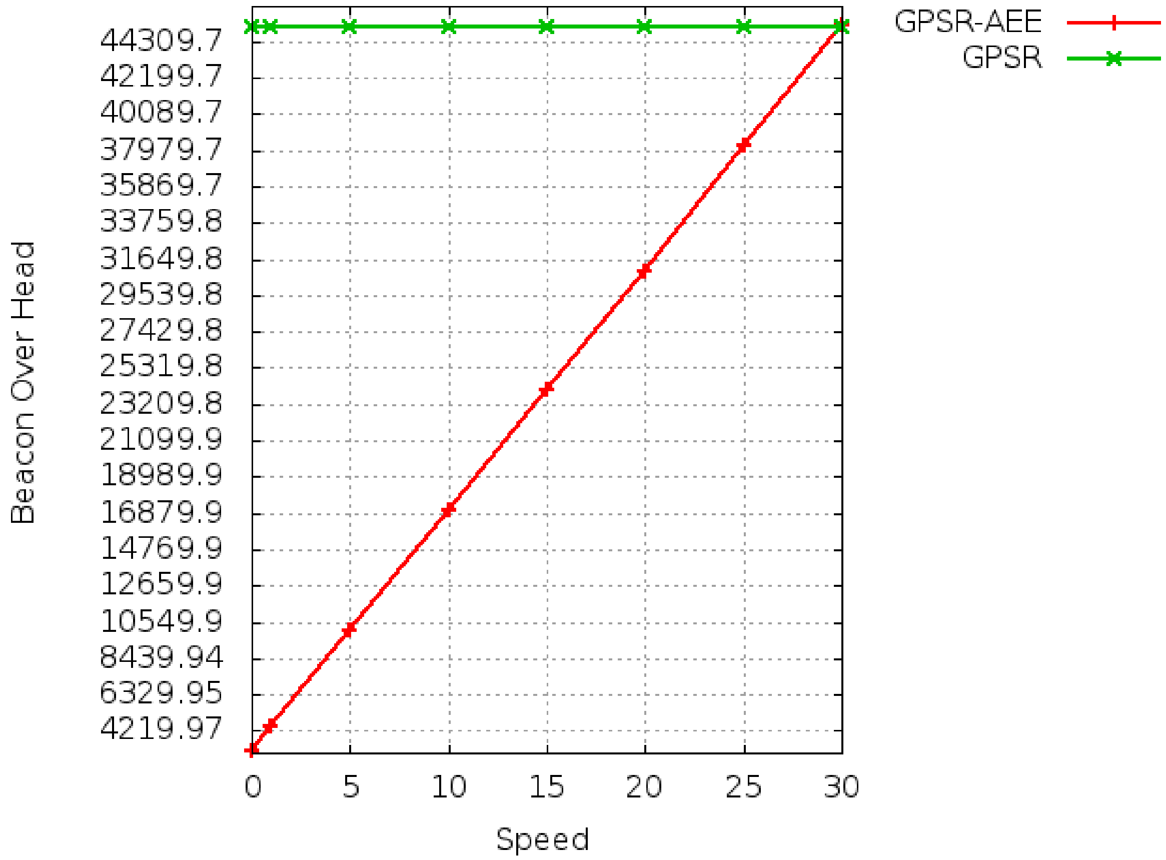

Figure 9. Simulation results showing effect of network mobility on Beacon Overhead

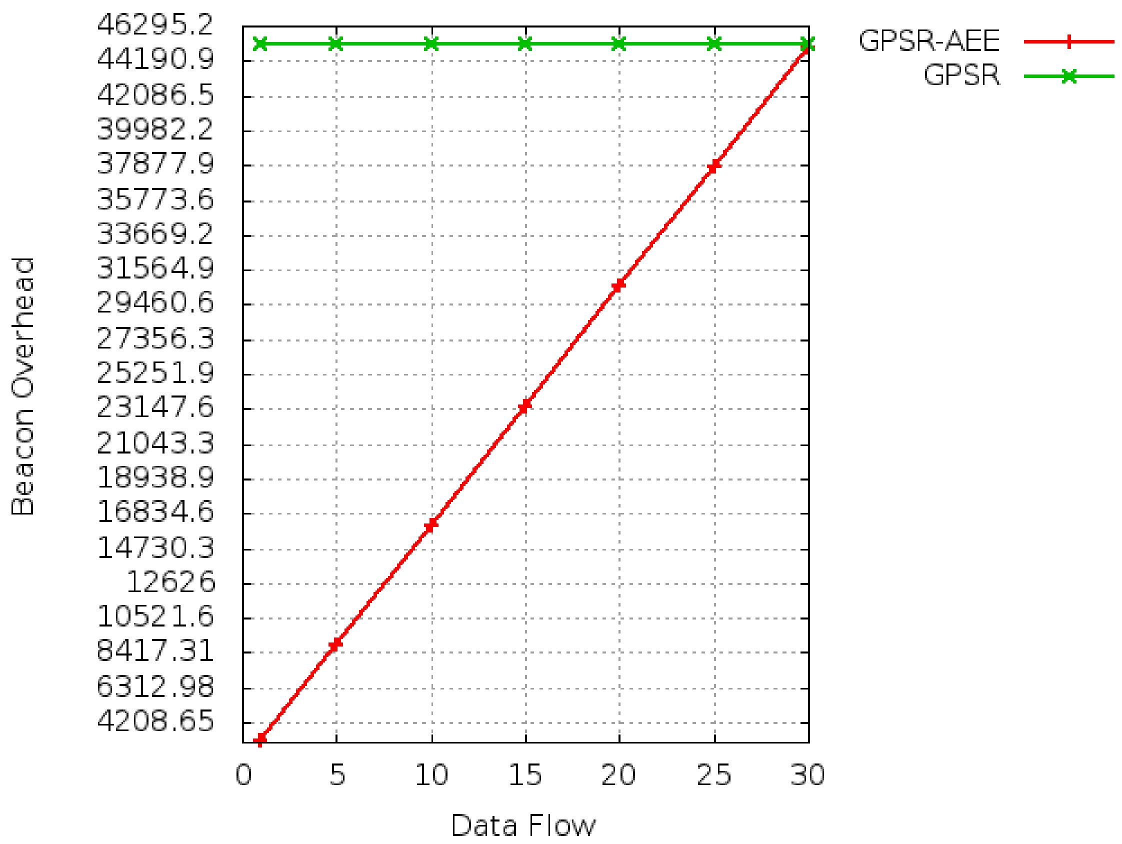

Figure 10. Simulation results showing effect of number of data flows on Beacon Overhead

Our mechanism (GPSR-AEE) is adjusting beacon interval frequently, while GPSR is using periodic beaconing mechanism. Therefore, GPSR is showing constantly high beacon overhead, while on the other hand our adaptive GPSR-AEE is showing drastically better performance. These results are depicted in Figures 9 and 10. Figure 9 illustrates the effect of network mobility on Beacon Overhead. Slower networks are showing up to 90 % lower beacon overhead than GPSR. A higher speed network produces more beacon overhead, as our mechanism is adapting to the network needs.

Figure 10 depicts the effect of number of data flows on Beacon Overhead. Lower data activity is also showing up to 90 % lower beacon overhead as GPSR-AEE sets low beaconing interval by taking the number of data flows into consideration. These graphs show that nodes in slow networks and networks with few data flows should set their beacon intervals to a longer period to generate fewer beacon updates (Figures 9 and 10), which results in lower beacon overhead. These results show the typical example of why the beaconing mechanism should be adaptive.

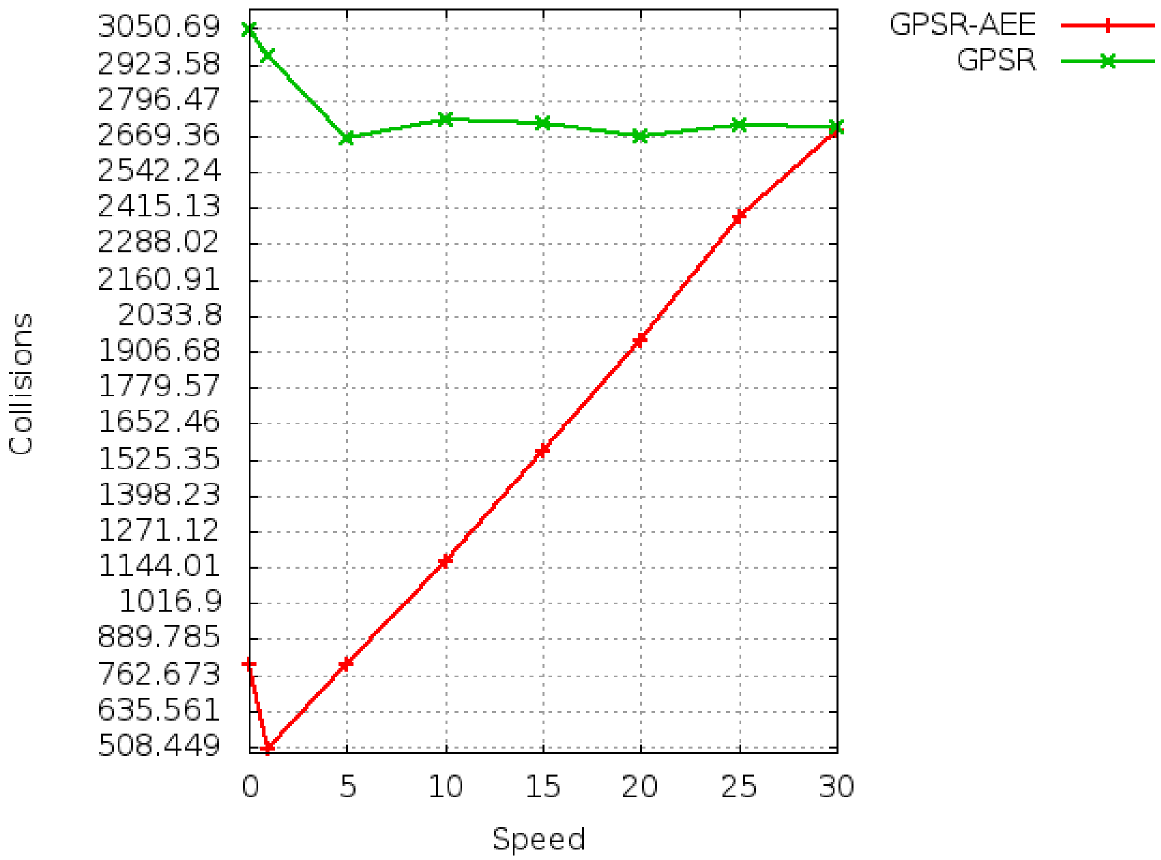

Figure 11. Simulation results showing effect of network mobility on number of packet collisions

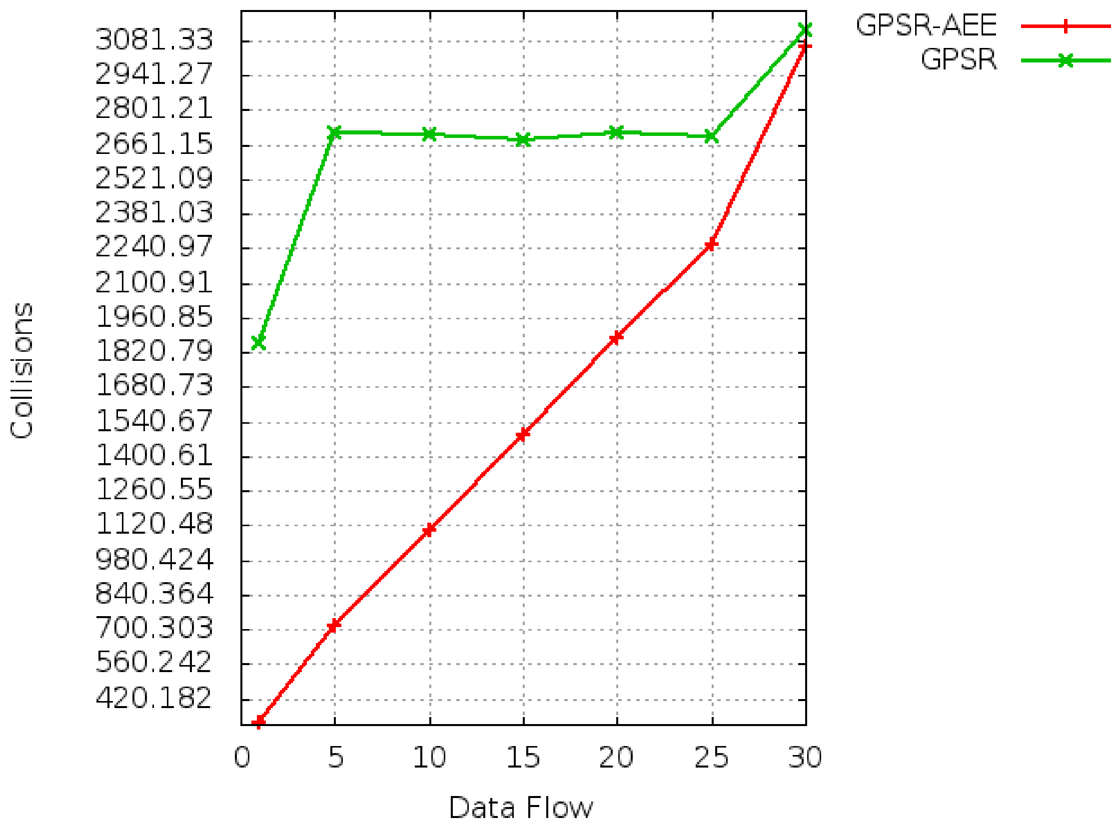

Figure 12 Simulation results showing effect of number of data flows on number of packet collisions

Figures 11 and 12 clearly show GPSR-AEE reduces MAC layer collisions. Generating smaller amount of beacon updates lowers the number of collisions for slow and moderately fast networks, as well as for networks with smaller numbers of data flows. For very high speeds and larger number of data flows collisions are inevitable as beacons require to be sent more often to preserve correct local topology for getting a high packet delivery fraction.

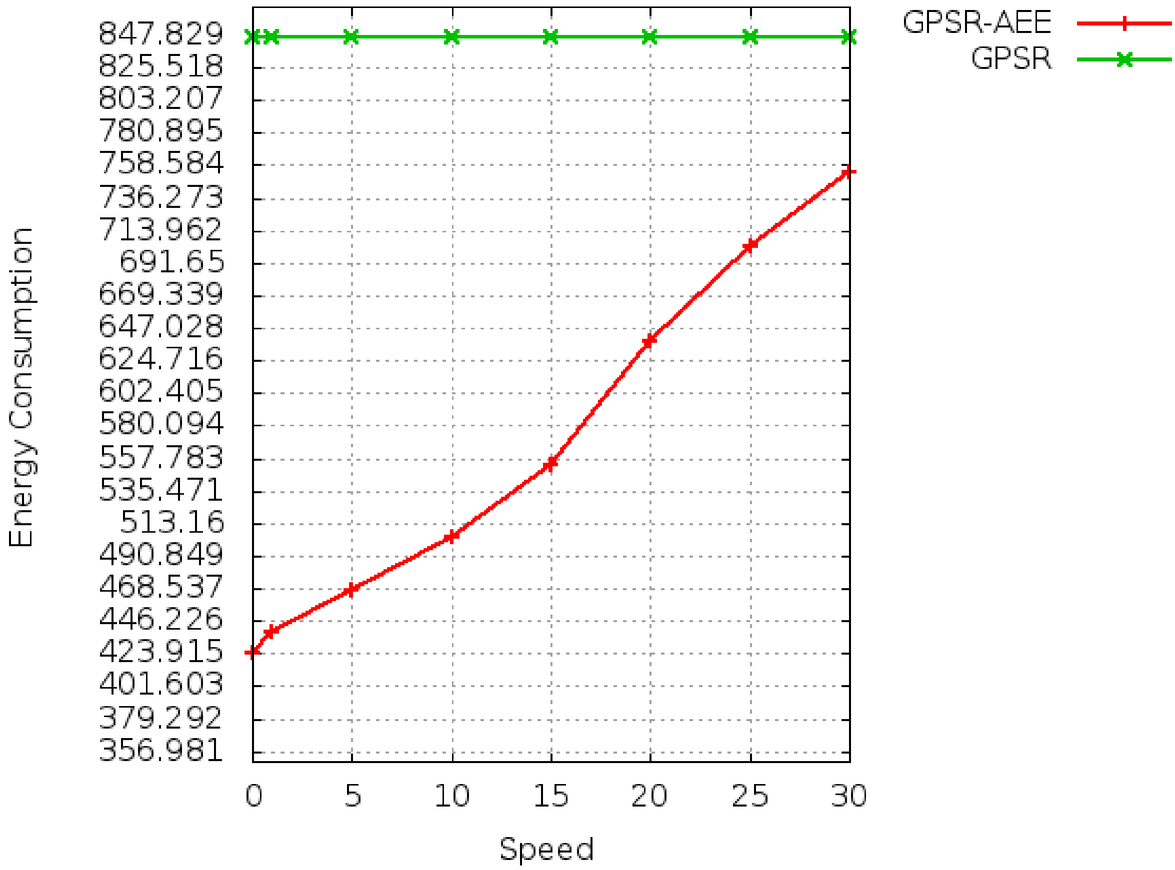

Figure 13. Simulation results showing effect of network mobility on Energy Consumption

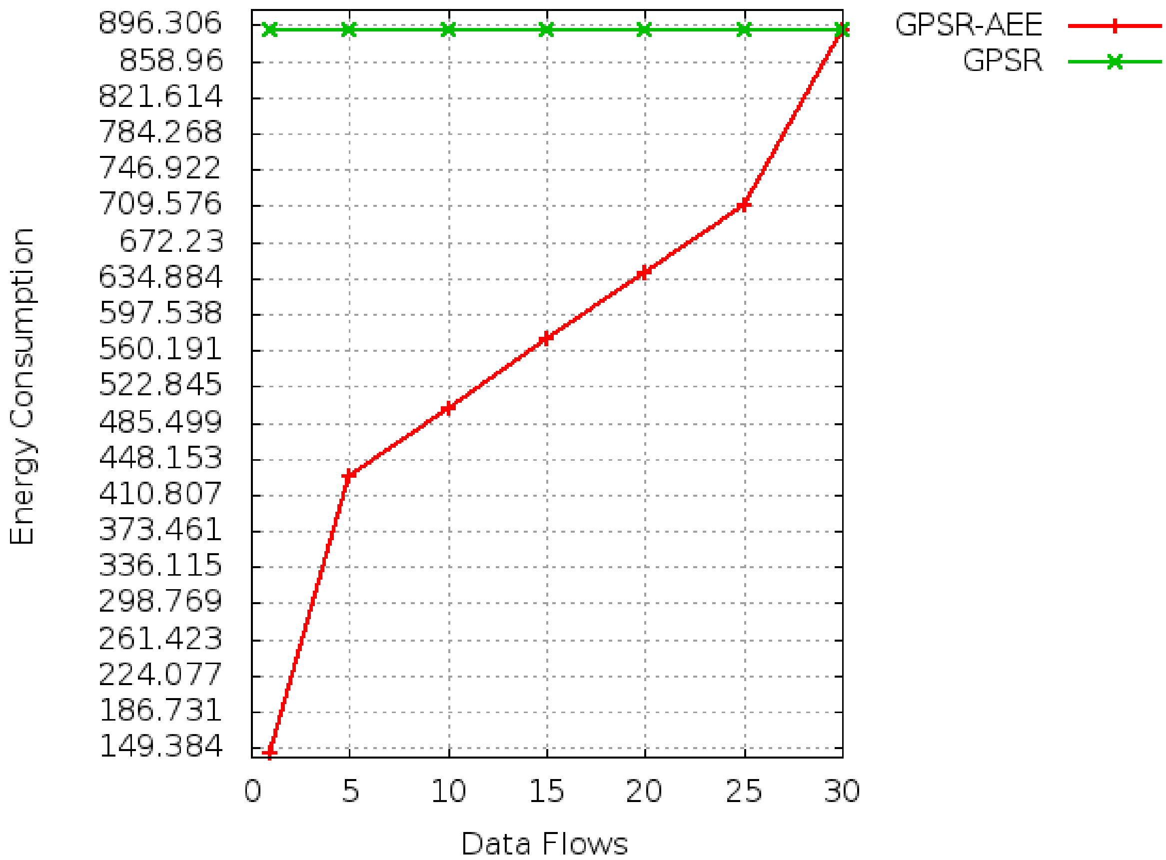

Figure 14. Simulation results showing effect of Number of Data Flows on Energy Consumption

Figures 13 and 14 show GPSR-AEE reduces the total energy consumption of the network. Beacon overhead, collision rate, energy consumption are all network metrics that are closely related to each other. So far we have noticed that generating smaller numbers of beacon updates reduces beacon overhead and collision rate. It also reduces the total energy consumption of nodes in the network. For low mobility energy consumption is reduces up to 50 %. Since energy is consumed to send packets as well as beacon updates, nodes generate fewer beacon packets and therefore use less energy. For low data load, GPSR-AEE reduces total energy consumption up to 83 %. By increasing speed and also by increasing the number of data flows, energy consumption raises because of more frequent beacon updates. However, energy consumed by GPSR-AEE is still significantly lower.

4.2 Results showing the impact of AEE on various node speed and various numbers of data flows with low initial energy

Table 2 shows the parameters and values used in this set of simulations:

Parameters	Value
Number of nodes	50
Topology	1500 m X 600 m
Duration of the simulation	900s
Transmission range	250 m
Maximum speed of the nodes	0~30 m/s (15m/s in simulation where we tested AEE for various numbers of data flows)
Number of Data Flows	1~30 (15 data flows in simulations where we tested AEE for various node speed)
Mobility model	Random Waypoint Model
Traffic Source	Constant Bit Rate (4 packets/second)
Data Packet Size	64 bytes
Initial energy level of each node	10 Joules

Table 3. Parameters of the second set of simulations

Figure 15~~,~~ shows the number of received packets in the network using GPSR with our energy-efficient adaptive mechanism GPSR-AEE and regular GPSR for various speeds of nodes. At the time when the nodes are static and slow, our mechanism is showing improved performance over GPSR. This is due to the higher value of beaconing interval, which means that nodes are still capable of maintain the accurate topology without creating unnecessary beacon overhead. For higher speeds, the number of received packets is lowered down. In this case, GPSR-AEE mechanism adapts beacon interval to smaller value so more frequent beacon updates are generated by nodes, which leads to higher energy consumption. Because of this, a lot of nodes are shutting down due to the power loss. However, our mechanism achieves higher numbers of received packets than the plain GPSR without adaptive beaconing mechanism. The reason for that is the adaptive nature of our mechanism. Fewer nodes are shutting down in the network using GPSR-AEE than in the network using GPSR (Figure 29). Higher speed indicates that more nodes need to maintain accurate topology of their neighbors.

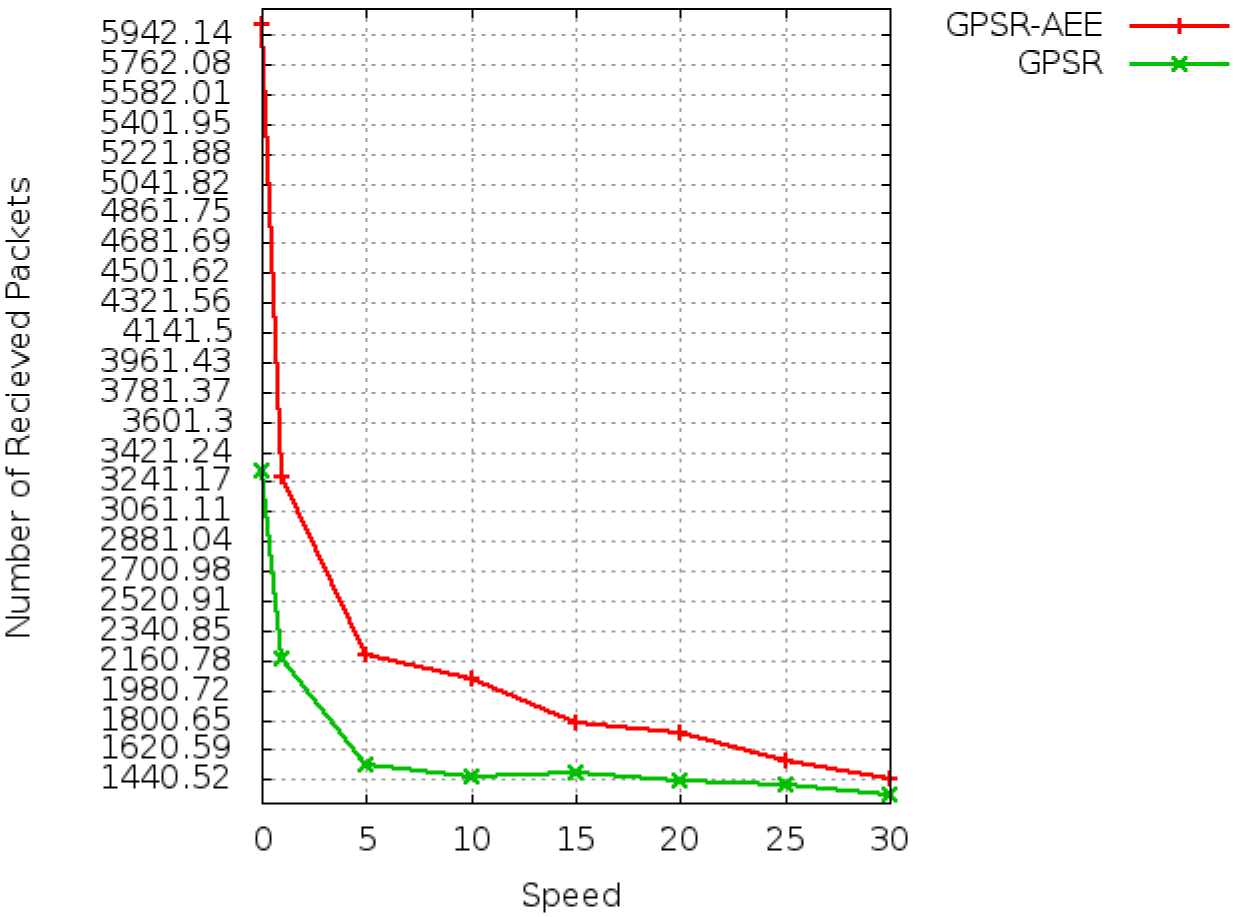

Figure 15. Simulation results showing effect of network mobility on number of received data packets

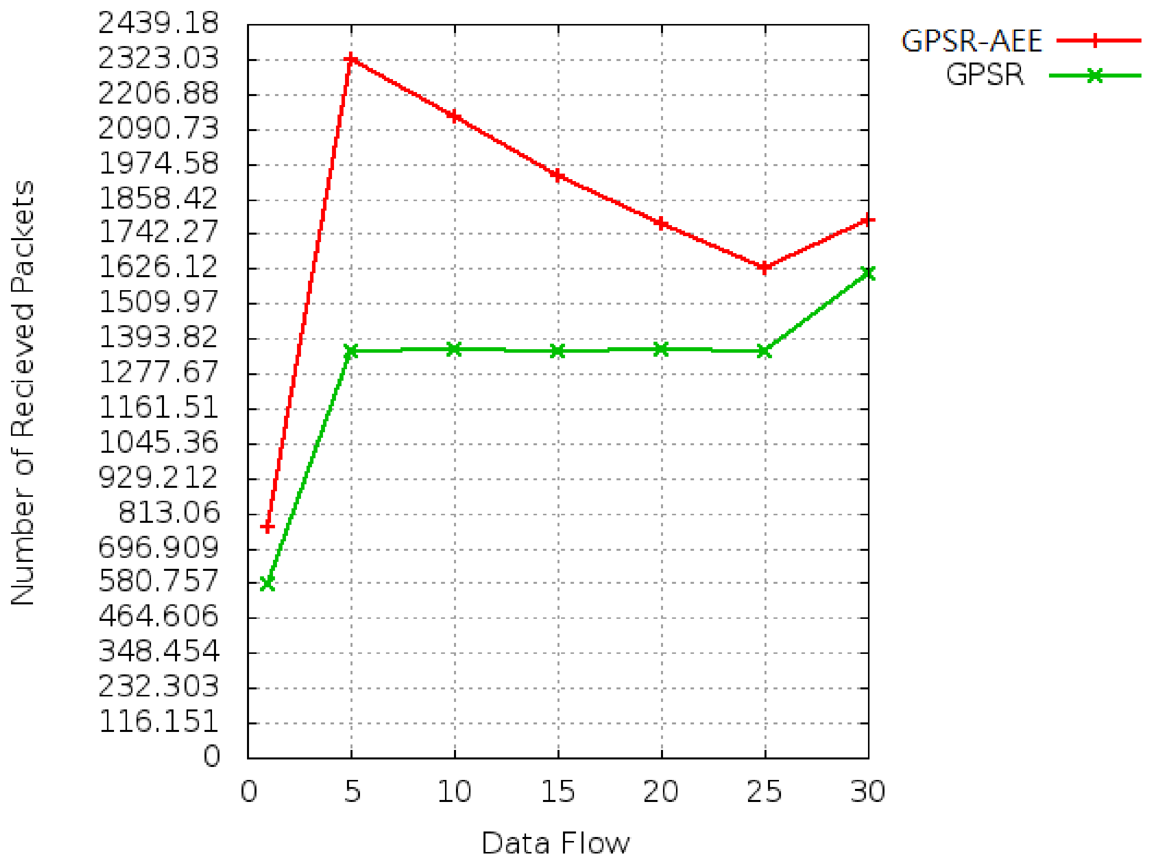

Figure 16. Simulation results showing effect of number of data flows on number of received data packets

Figure 16 depicts the effect of number of data flows on number of received data packets. In networks with one data flow GPSR-AEE achieves higher number of received packets by the destination node. This metric is drastically higher for networks with 5 data flows because of the increased traffic load. The number of received packets of GPSR in networks of 5 to 25 data flows is constant. This is due to the number of shut nodes depicted in Figure 30. In this situation nodes are shutting almost at a constant pace, which prevents transmission and receiving of packets. The number of received packets of GPSR-AEE in networks of 5 to 25 data flows is linearly decreasing as the number of shut nodes is linearly increasing (Figure 30). For a very high number of data flows (30) the number of received data packets is increased due to 5 extra source destination pairs that are generating and receiving packets. Accurate topology is maintained while saving a lot of energy. One of the main reasons for a higher number of received packets is the reduced energy consumption (Figure 27 and 28) of GPSR-AEE as many nodes in the network that are using GPSR are shutting down due to the energy loss. Therefore they are unable to transmit data.

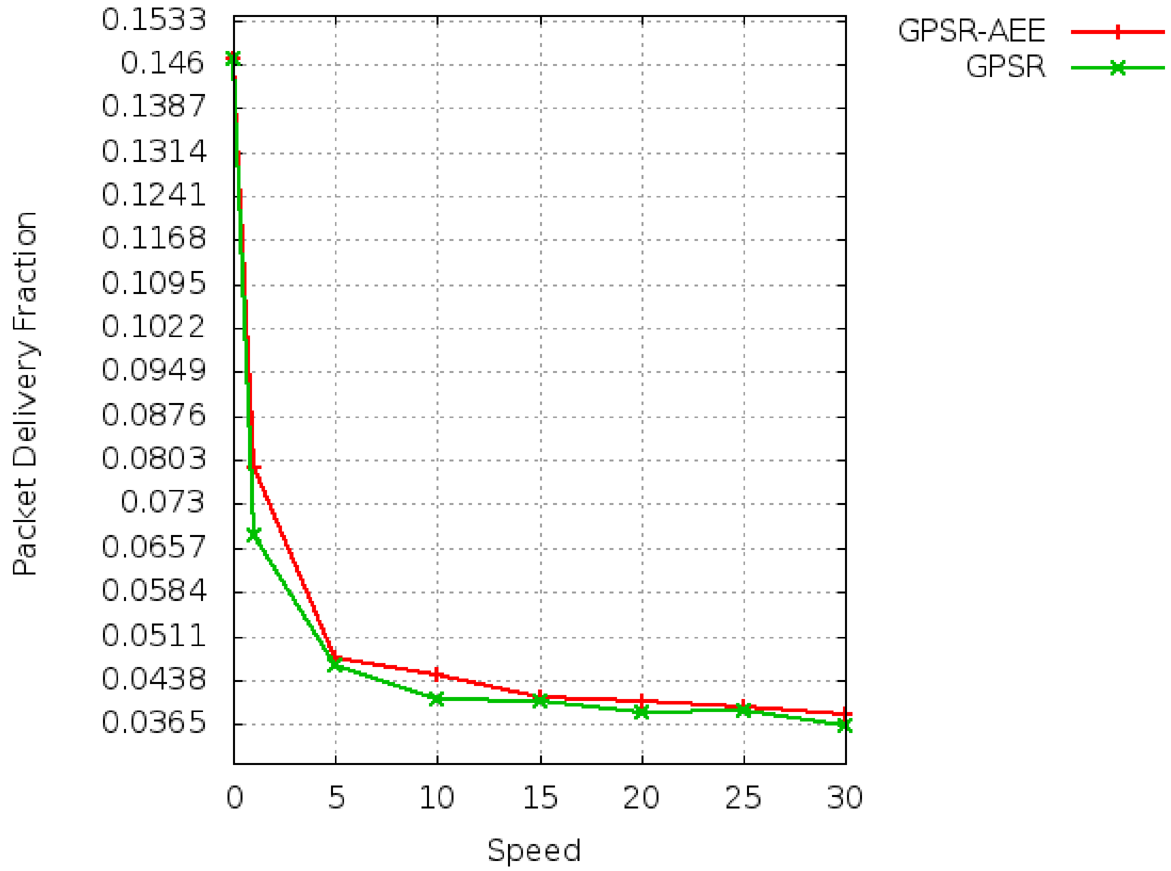

Figure 17. Simulation results showing effect of network mobility on Packet Delivery Fraction

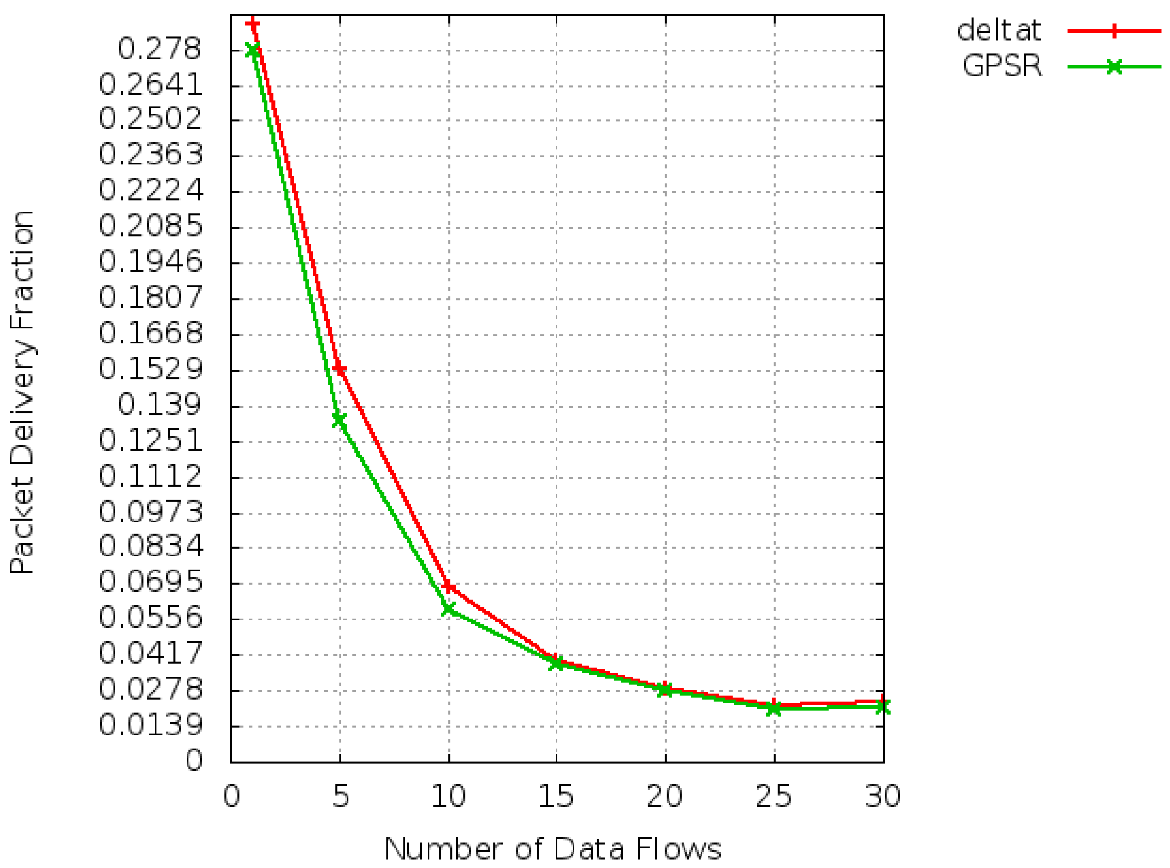

Figure 18. Simulation results showing effect of number of data flows on Packet Delivery Fraction

Not only was the number of received packets positively influenced by GPSR-AEE, packet delivery fraction is as well. Static nodes are showing better performance judging by Figure 17 and 18. But these results can't be looked individually. If we look at Figure 15 again, we can see that the number of received packets when the speed is equal to zero is twice larger for GPSR-AEE than GPSR. Saying this, we can conclude that in this case the number of received packets is a more vital metric than the packet delivery fraction. We can also see that GPSR-AEE is improving packet delivery fraction for mobile nodes.

The average hop-count is significantly lowered comparing to plain GPSR. More accurate topology is being maintained with the AEE mechanism, so the routing path is closer to optimal. Figure 19 and Figure 20 show that the number of nodes which are shutting down due to energy losses is drastically smaller using our mechanism. This provides the possibility for multiple options when making routing decisions, leading to lower hop count metric.

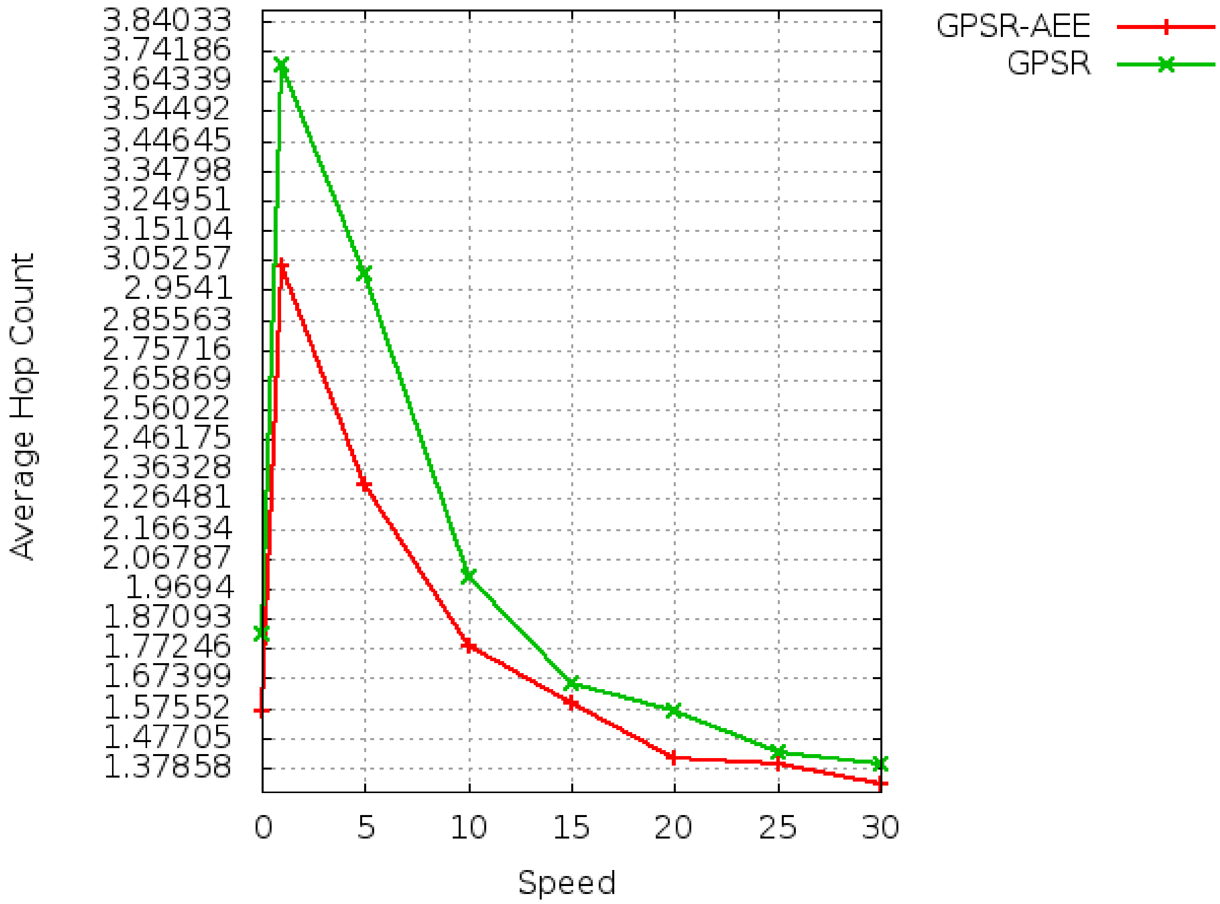

Figure 19. Simulation results showing effect of network mobility on Average Hop Count

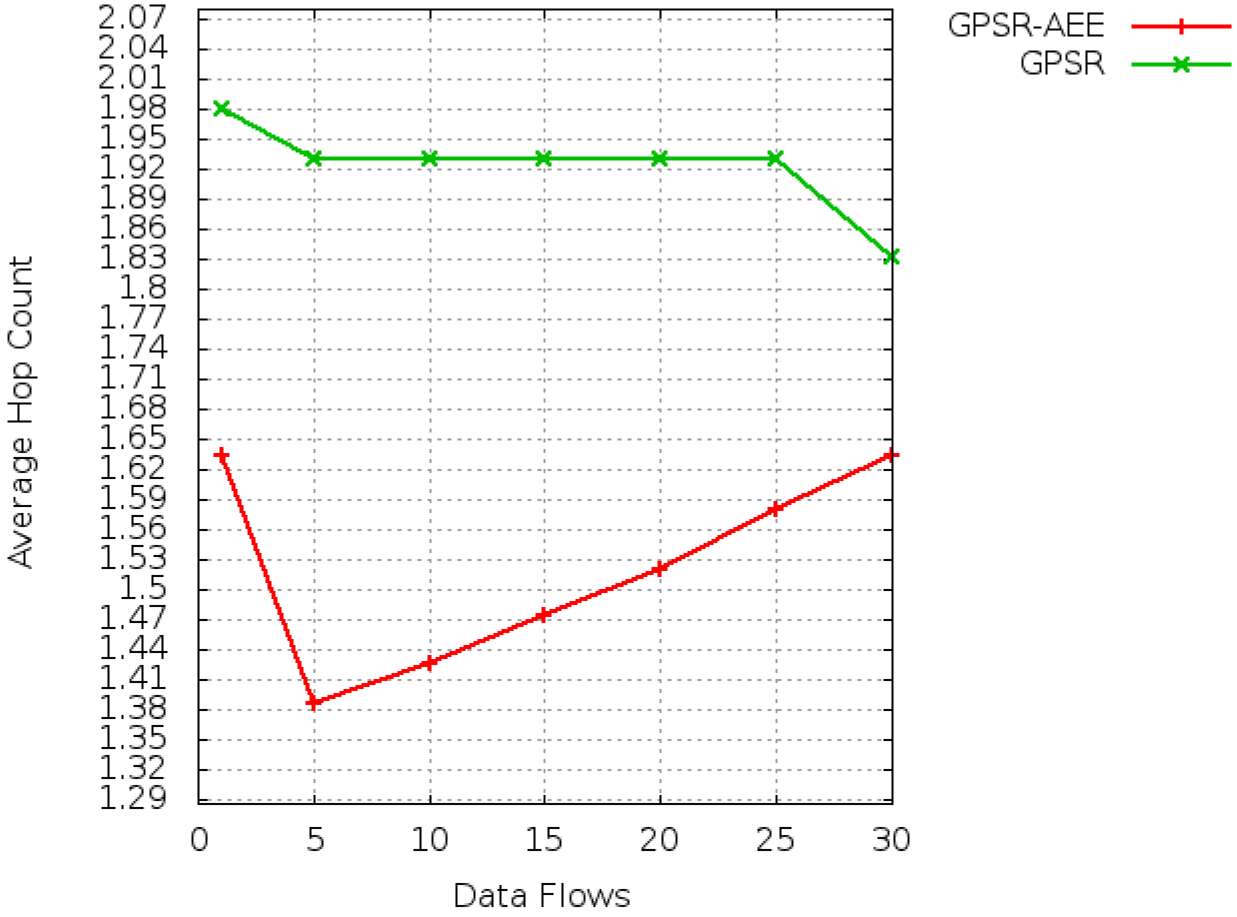

Figure 20. Simulation results showing effect of number of data flows on Average Hop Count

Average End-to-end Delay (Figure 21 and 22) and hop count should show some similarity in results as both of the metrics are mutually dependent. As mentioned before, our mechanism is managing more accurate topology of the network, thus achieving results of lower hop count metric. Lower hop-count metric of GPSR-AEE leads to lower End-to-end Delay than GPSR as packets are traversing fewer nodes. Hop-count represents the measurement of distance in a network, therefore the longer the “distance” that packets need to take the longer the End-to-end Delay.

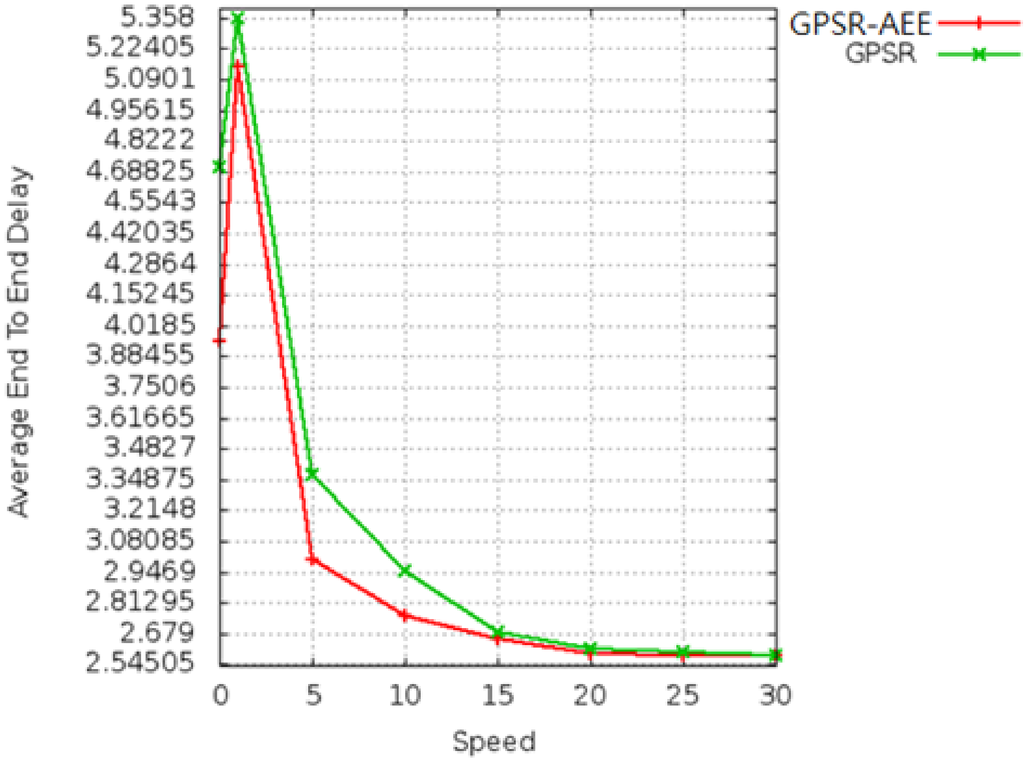

Figure 21. Simulation results showing effect of network mobility on Average End to end Delay

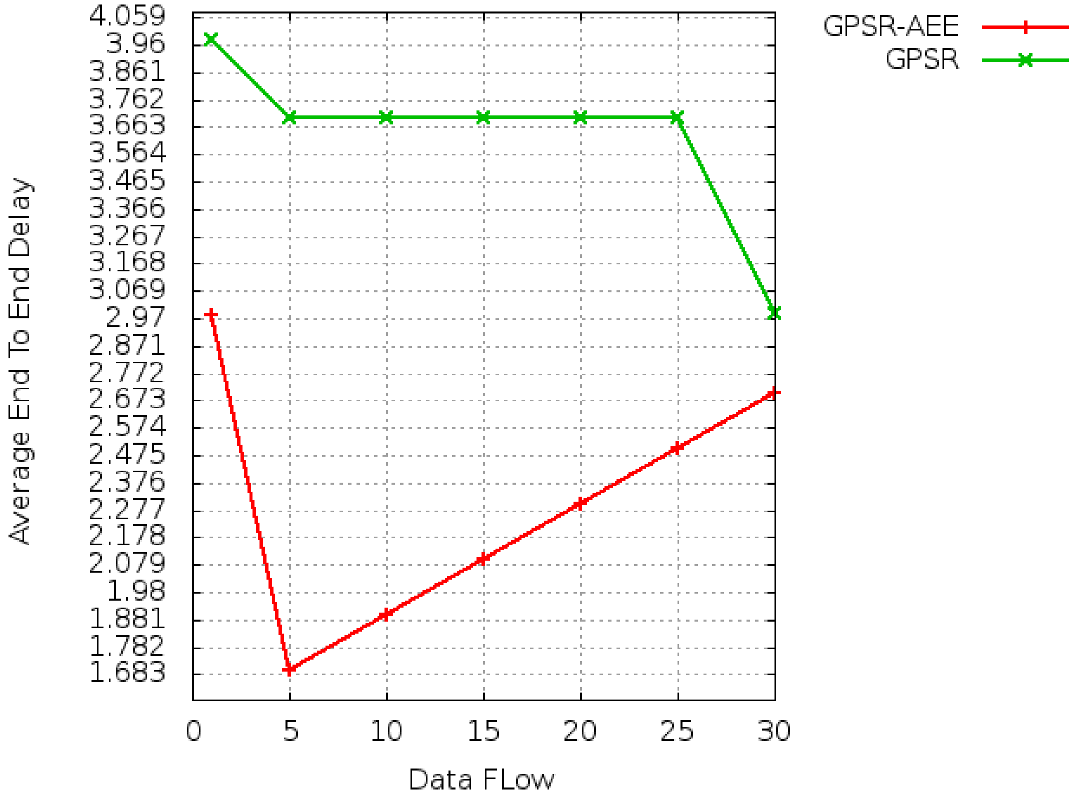

Figure 22. Simulation results showing effect of number of data flows on Average End to end Delay

GPSR beaconing mechanism is strictly periodic, for that reason beacon overhead is constant if nodes would never run out of energy. However, when nodes are shuting down due to energy loses beacon overhead of GPSR is changing as we can se in figures 23 and 24. Nodes that have ran out of energy are unable to send becaon updates, therefore beacon overhead is lower. GPSR-AEE is poducing less beacon overhead because it is adapting its beacon interval to network mobility and traffic load. For low mobility GPSR-AEE reduces beacon overhead up to 88 %. This is due to the fact that for low mobility beaconing interval is set to a longer time , therefore nodes generate fewer beacon updates. For higher mobility GSPR-AEE sets the beacon interval to a shorter value in order to maintain precise ~~t~~local topology.

For smaller numbers of data flows GPSR-AEE diminishes beacon overhead up to 94%. Beacon overhead of GPSR-AEE is increasing linearly with the increased number of data flows.

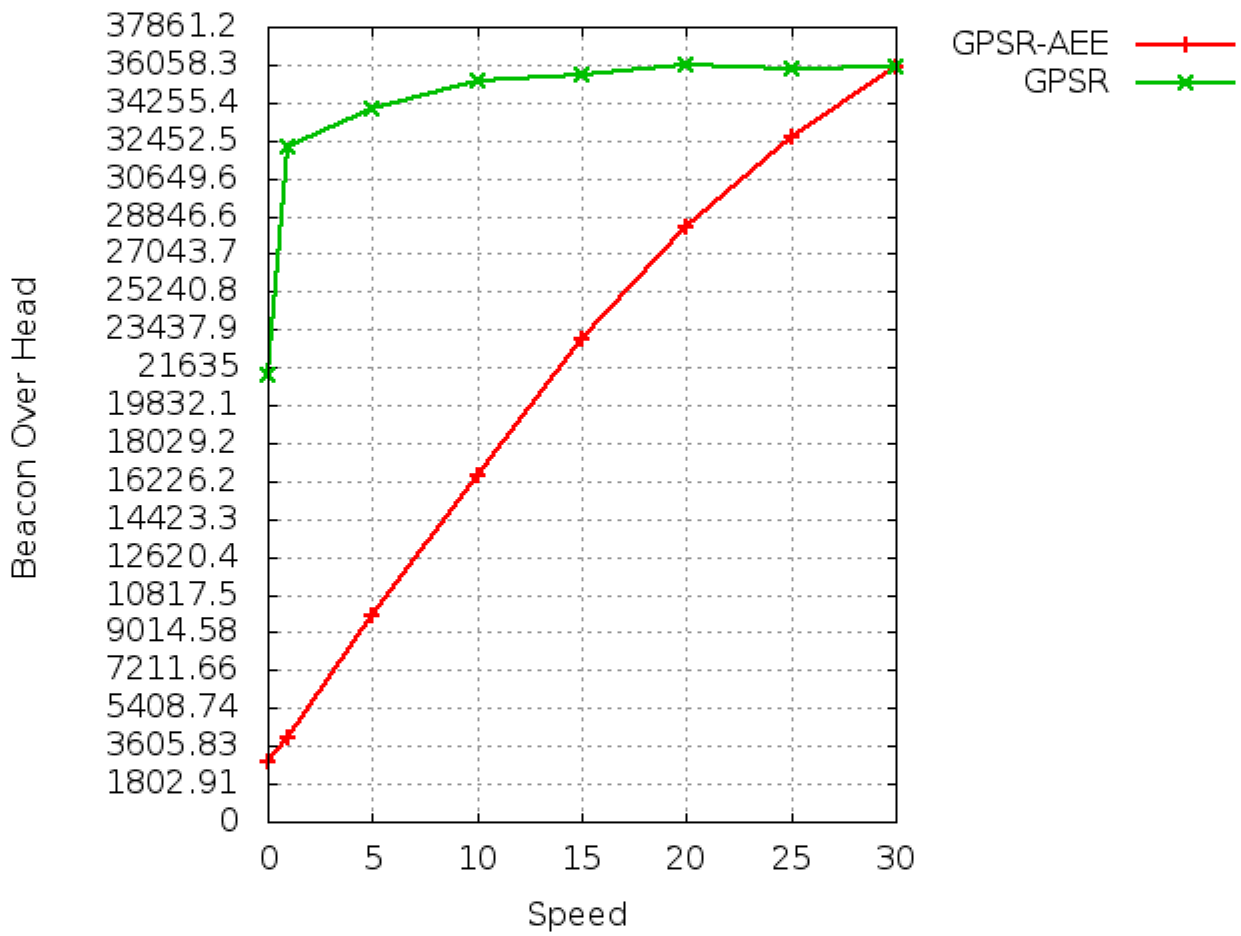

Figure 23. Simulation results showing effect of network mobility on Beacon Overhead

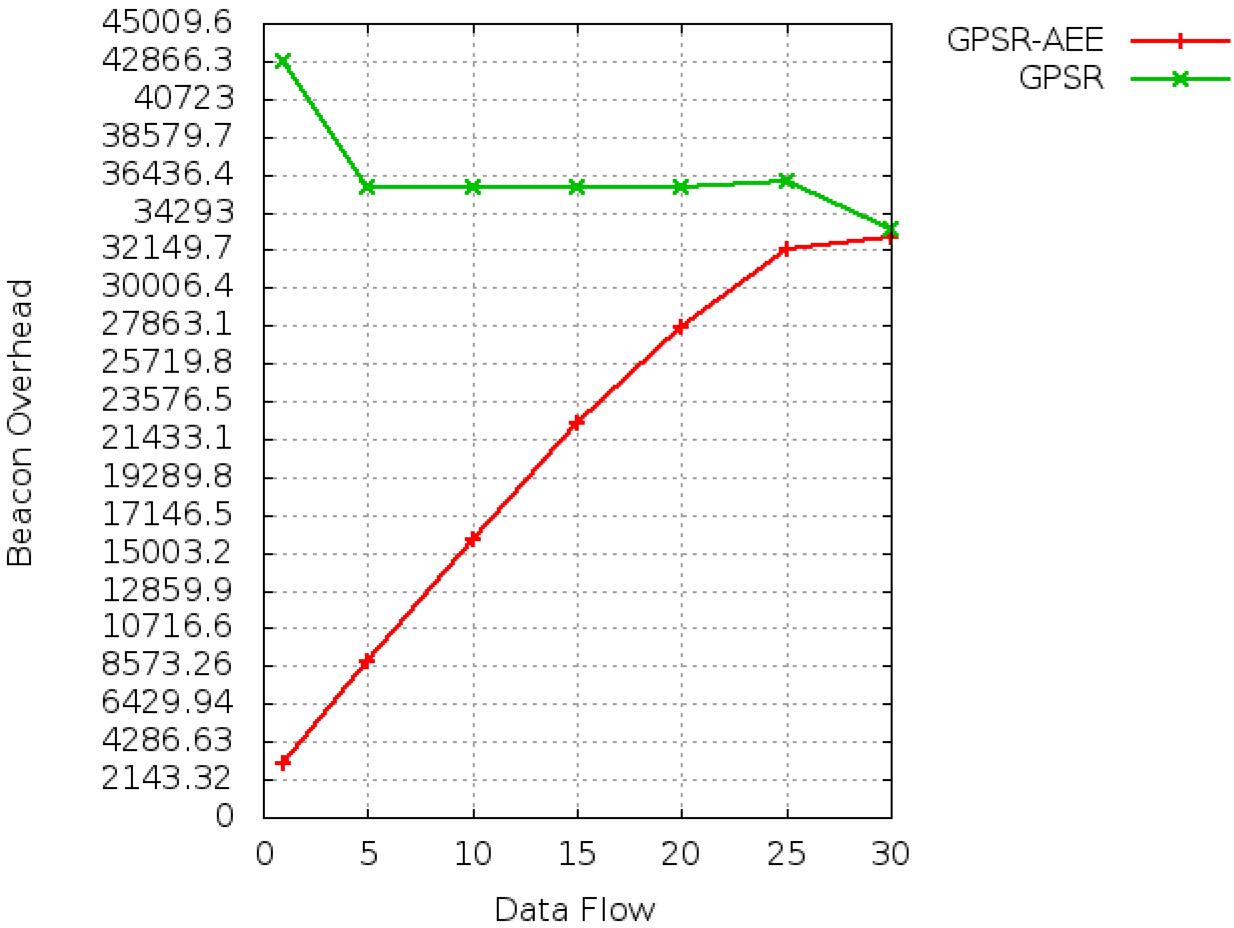

Figure 24. Simulation results showing effect of number of data flows on Beacon Overhead

A reduced amount of beacon updates generates less collision in MAC layer. For higher traffic load and higher speed of the network GPSR-AEE sets beaconing interval to shorter value in order to maintian more accurate local topology. Beacon updates are then more frequent so more collisons occur. But still, GPSR-AEE produces up to 70% fewer collisions than GPSR as we can see in Figure 25. GPSR-AEE also results in up to 77% fewer collision than GPSR as we can see in Figure 26.

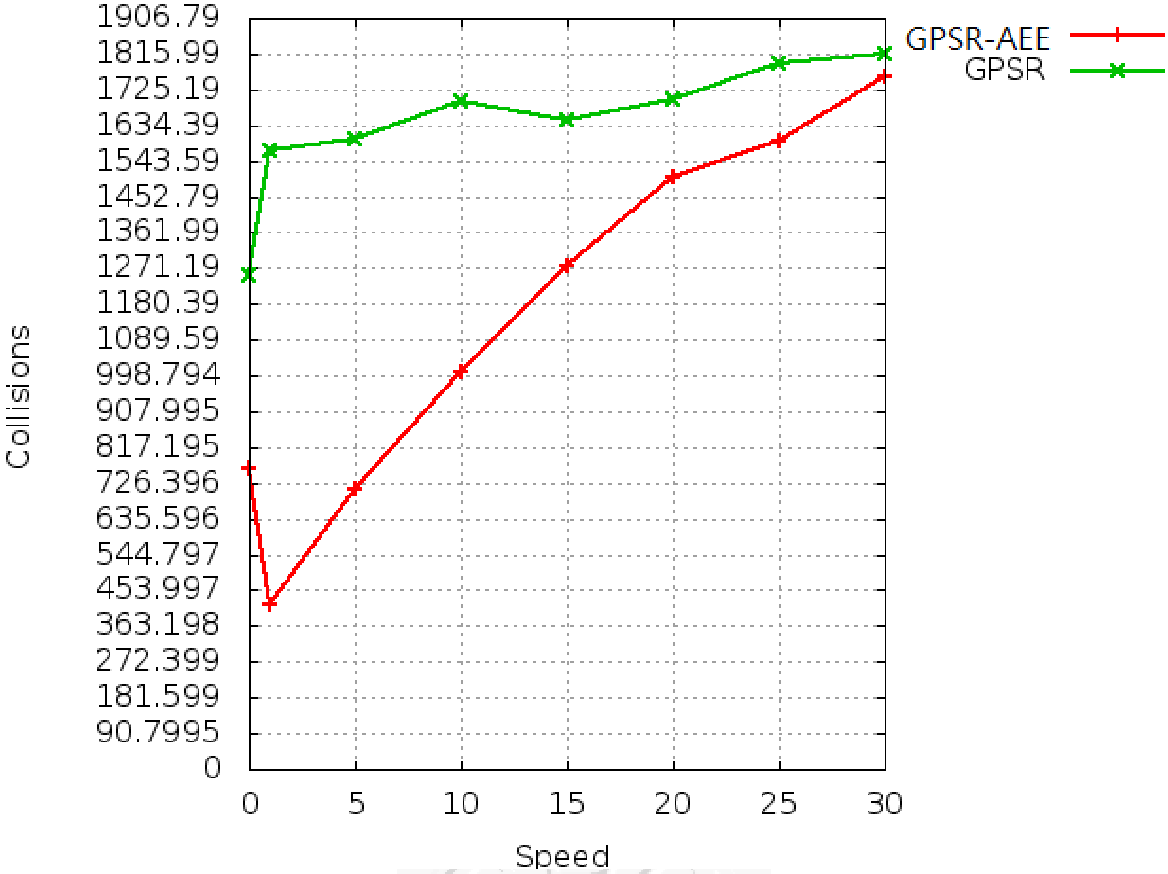

Figure 25. **Simulation results showing effect of network mobility on number of packet collisions**

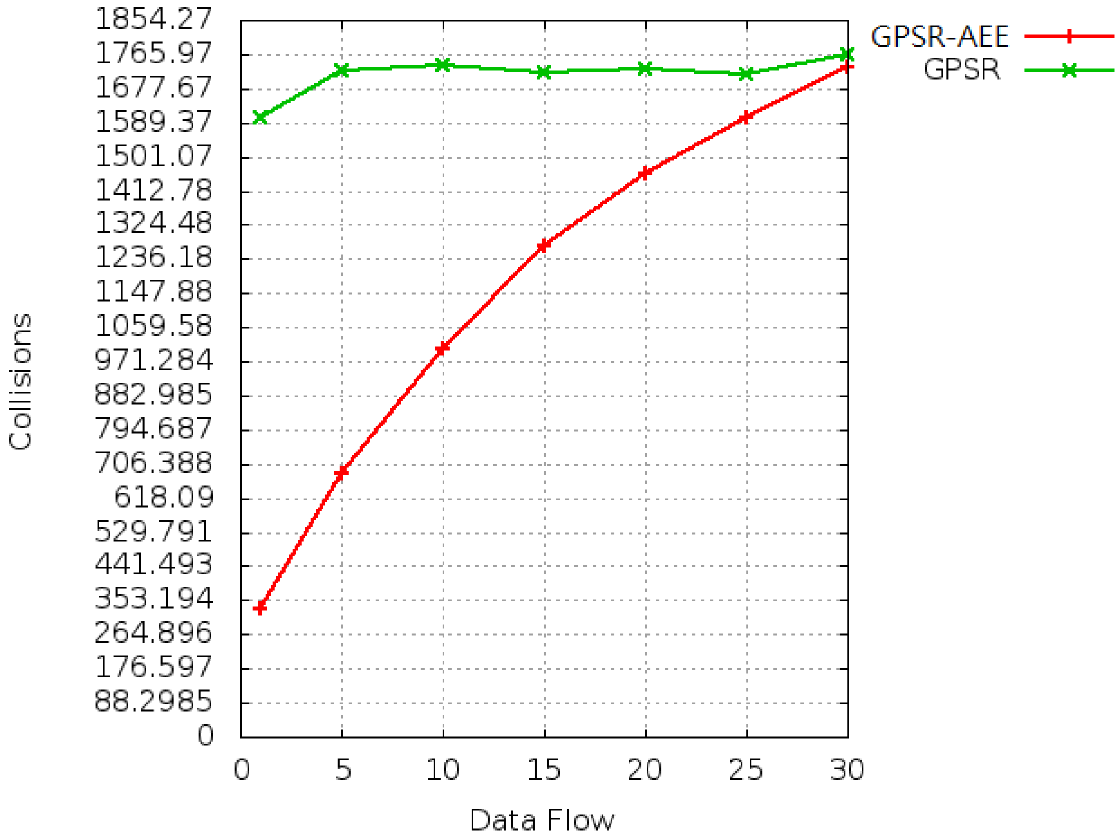

Figure 26. **Simulation results showing effect of number of data flows on number of packet collisions**

Energy consuption is lower using GPSR-AEE in slow and moderatly fast networks as well in the networks with lower number of data flows (Figures 27 and 28). For high speed and higher number of data flows, higher energy consumpiton is inevediable because of GPSR-AEE adapting nature.This not only saves total energy consumed, it also manages to reduce the number of powered off nodes.

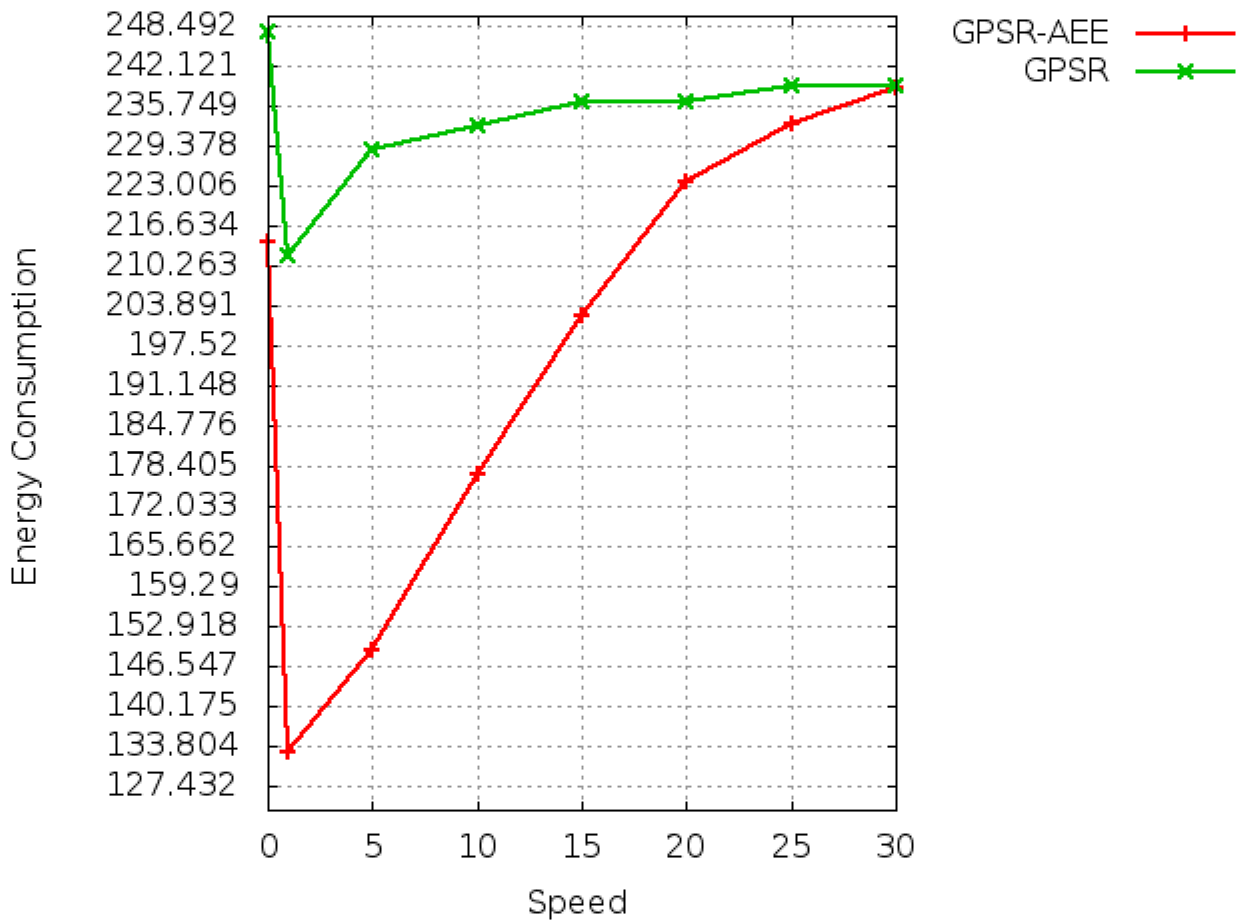

Figure 27. Simulation results showing effect of network mobility on Energy Consumption

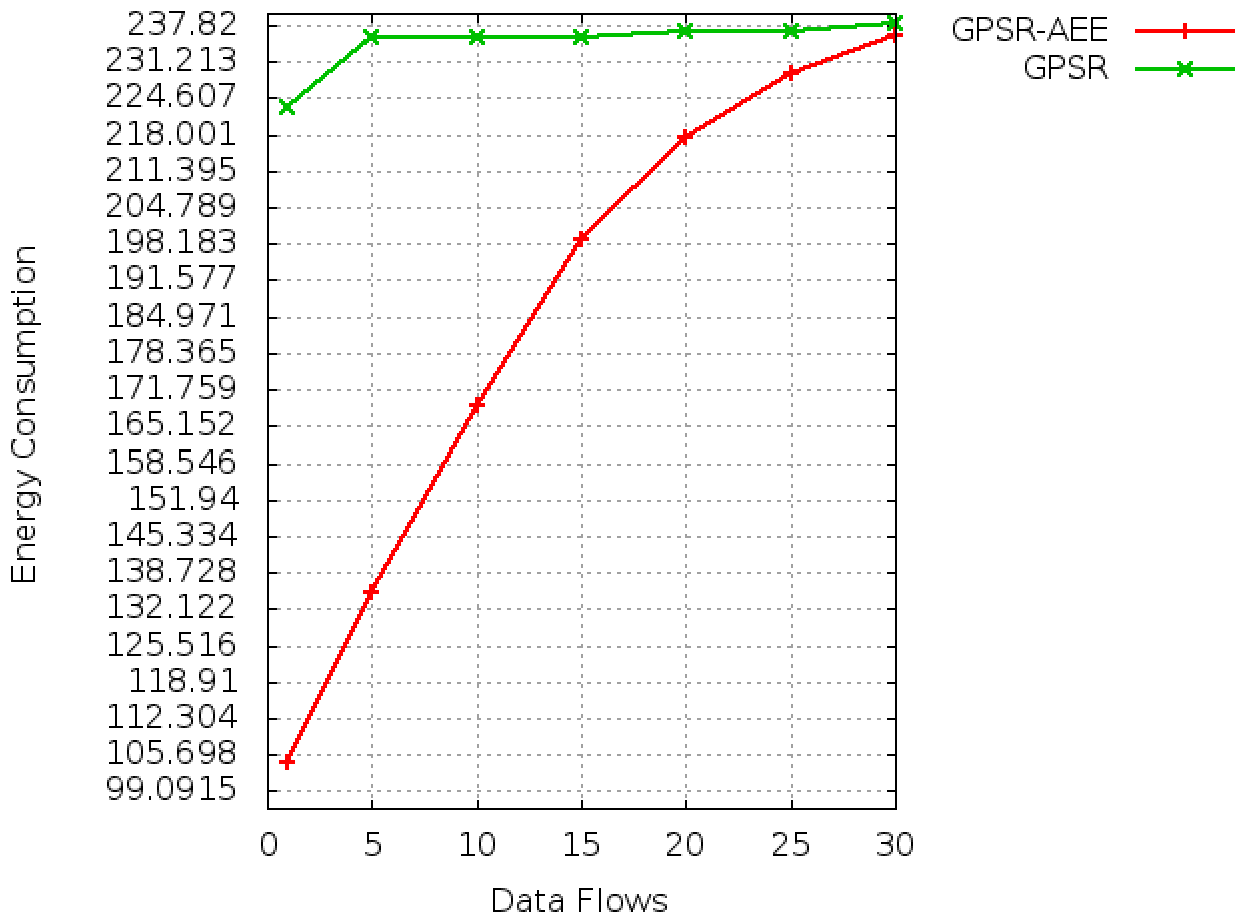

Figure 28. Simulation results showing effect of Number of Data Flows on Energy Consumption

Powered off or shut nodes can't transmit data, which thus lowers the total number of received packets by destination nodes. The number of shut nodes for slow and moderalty fast nodes is almost around six times smaller using GPSR-AEE comparing to GPSR in Figure 29. Similar performance is shown in Figure 30 as for a low number of data flows around 13 times less nodes are powering off.

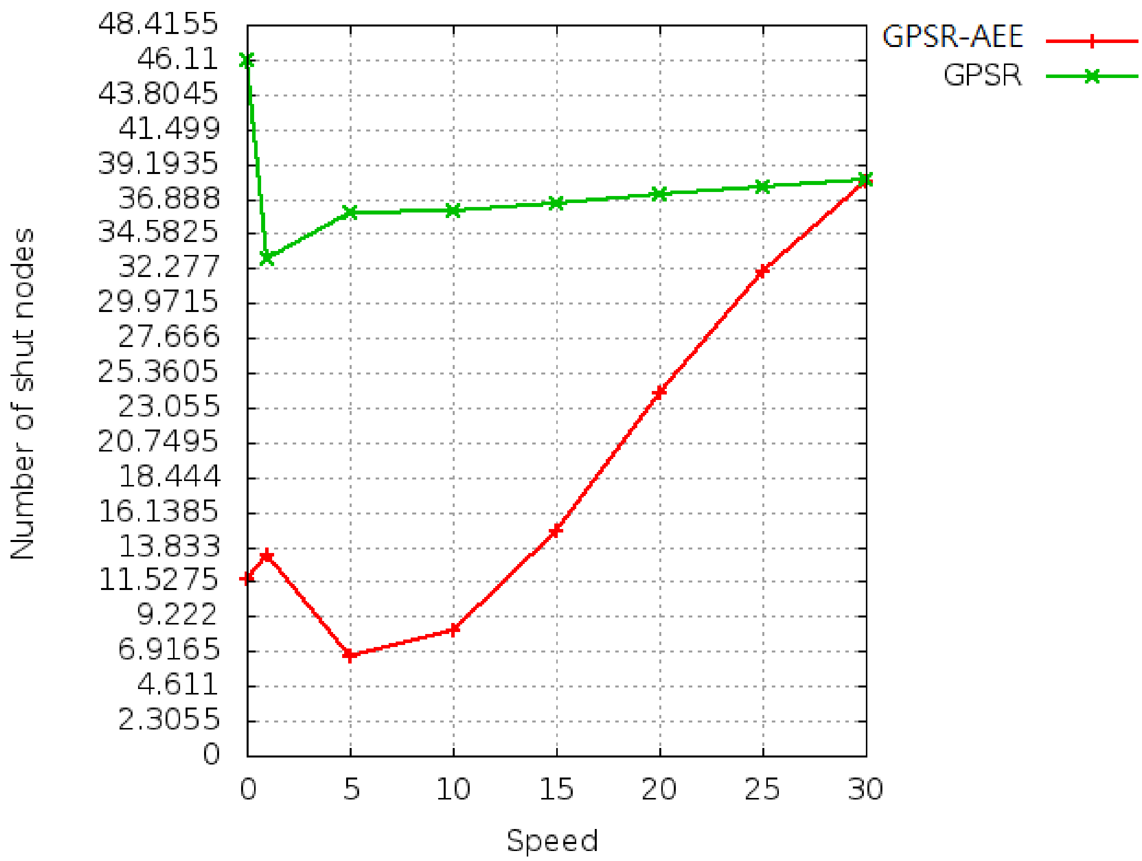

Figure 29. **Simulation results showing effect of network mobility on number of shut nodes**

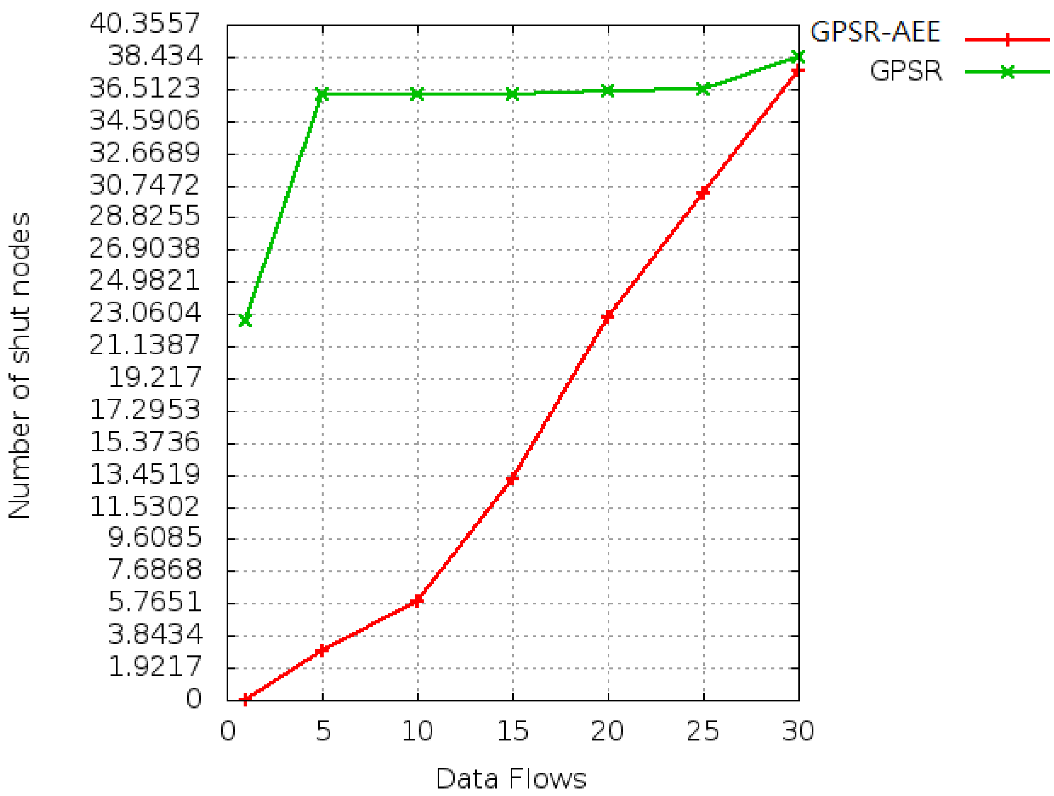

Figure 30. **Simulation results showing effect of number of data flows on number of shut nodes**

Figures 31 and 32 shows the nodes which are powering off in reference to simulation time. We can see that nodes using GPSR are losing energy earlier in the simulation and therfore quite a larger number of them is shutting down for both slow and fast mobility. This is happening due to an unnecessary high amount of beacon updates that GPSR generates, therefore lot of nodes consumes all of its energy earlier comparing to GPSR-AEE.

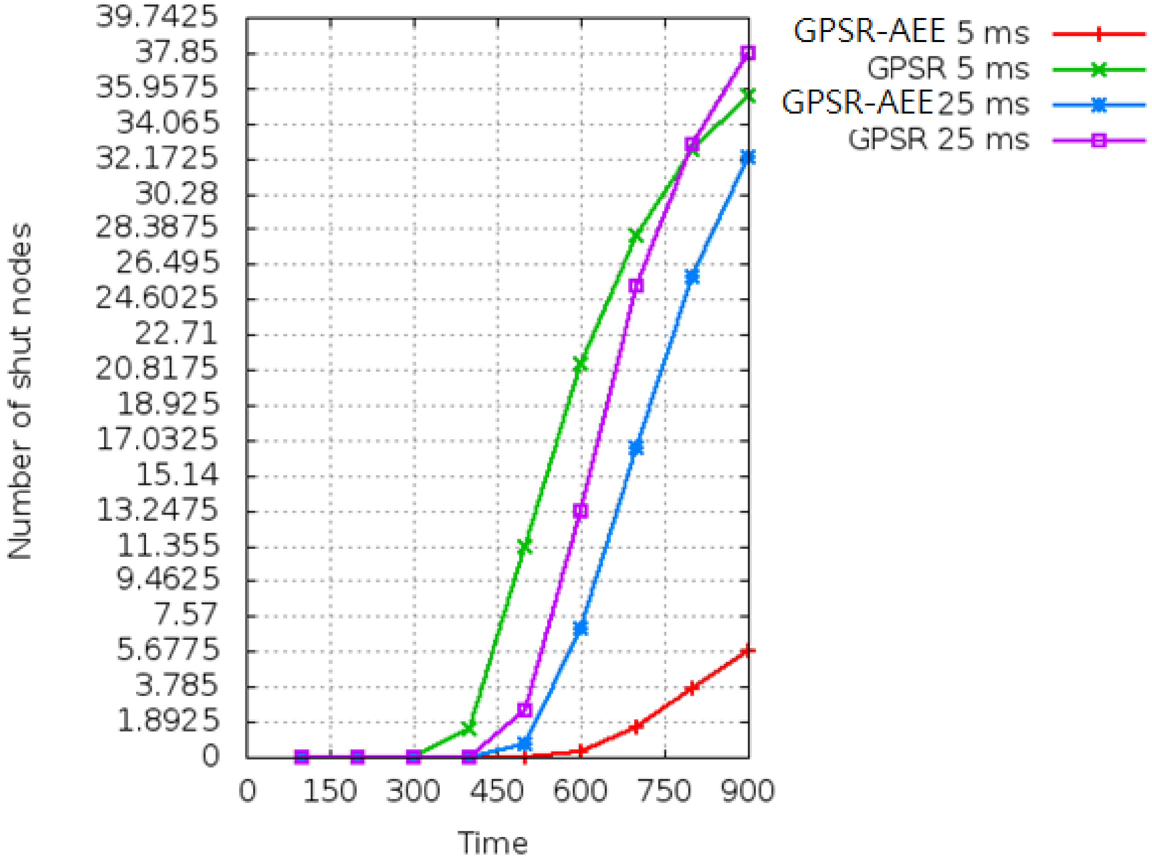

Figure 31. Simulation results showing effect of network mobility on number of shut nodes throughout the simulation time

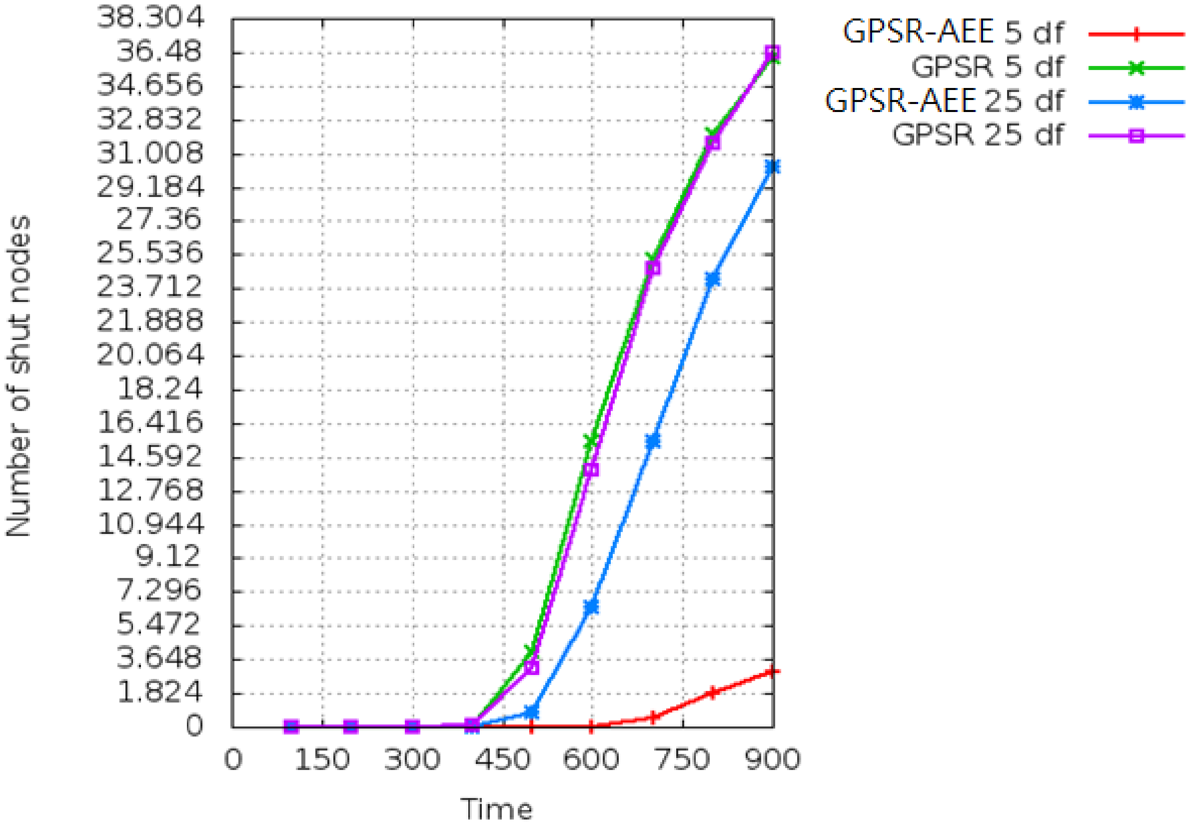

Figure 32. Simulation results showing effect of number of data flows on number of shut nodes throughout the simulation time

5. Conclusion

We have demonstrated the necessity to adjust the beacon interval used in geographic routing depending on mobility and traffic load of the network. Our Adaptive Energy-Efficient (AEE) beaconing mechanism deals with these problems. Beaconing interval is adapted based on the speed of the node as well as on the number of data flows traversing to each individual node. This method allows nodes to generate more beacons in case of high mobility and high data transmission. Furthermore, by letting nodes generate fewer beacons in case of low mobility and low data transmission the beacon overhead, energy consumption and collision rate all become lower.

We have inserted AEE within GPSR and tested its performance in comparison with regular GPSR using comprehensive ns-2 simulations. Results are showing numerous improvements to network metrics. The number of beacon updates decreases while the packet delivery fraction and number of received packets increases. Additionally, energy consumption and collision rate are drastically lowered down comparing to regular GPSR. “Life” of the network is longer with GPSR-AEE, as nodes are having longer life-time to transmit data before losing all energy. Since our mechanism (GPSR-AEE) is maintaining more accurate topology of the network hence improvement in hop-count and end to end delay is achieved.

References

[1] Y. Wang, T. Wu, W.lee, and C.Ke, "A novel geographic routing srategy over VANET," *In 24th International Conference on advanced Information Networking and Applications Workshops (WAINA)*, April 2010

[2] Chai K Toh, " Ad Hoc Mobile Wireless Networks: Protocols and Systems" 2002

[3] Tomas Krag and Sebastian Büettrich (2004-01-24). "Wireless Mesh Networking". *O'Reilly Wireless Dev Center*. Retrieved 2009-01-20.

[4] B. Karp and H. T. Kung, "Greedy perimeter stateless routing for wireless networks," in *ACM/IEEE MOBICOM*, August 2000, pp. 243-254.

[5] Q. Chen"Analysis and Application of Hop Count in Multi-hop Wireless Ad-hoc Networks" Department of Computer Science and Engineering University of New South Wales 2009

[6] X. Xiang , Z. Zhou and X. Wang "Self-Adaptive On Demand Geographic Routing Protocols for Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2007

[7] Y. Jia, W. Ying-you and Z. Hong "A solution to outdated neighbor problem in wireless sensor networks," (2010) *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 5, art. no. 5563956, pp. 277-280.

[8] Q. Chen, S. S. Kanhere, and M. Hassan, "Adaptive Position Update for Geographic Routing in Mobile Ad-hoc Networks," *IEEE Transactions on mobile computing*, vol. 12, no. 3, March 2013

[9] The VINT Project. The UCB/LBNL/VINT Network Simulator-ns 2.http://www.isi.edu/nsnam/ns/

[10] X. Li, J. Yang, A. Nayak and I. Stojmenovic, Localized Geographic Routing to a Mobile Sink with Guaranteed Delivery in Sensor Networks,&rdquo, *IEEE J. Selected Areas in Comm.*, vol. 30, no. 9, pp. 1719-1729, Oct. 2012.

[11] H. de Oliveira , A. Boukerche , E. Nakamura and A. Loureiro "An efficient directed localization recursion protocol for wireless sensor networks", *IEEE Trans. Comput.*, vol. 58, no. 5, pp.677 -691 2009

[12] F.C. Yu, Y.H. Choi, S.H. Park, E.S. Lee, M.S. Jin, and S.H. Kim, "Sink Location Service for Geographic Routing in Wireless Sensor Networks", *IEEE Wireless Communications & Networking Conference(WCNC 2008)*, March, 2008, Las Vegas, USA

[13] C. Perkins, E. Belding-Royer and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*,IETF RFC 3561, July 2003

[14] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker and I. Stoica, "Geographic Routing without Location Information,&rdquo", *Proc. ACM MobiCom*, pp. 96-108, Sept. 2003

[15] Q. Chen, S.S. Kanhere, M. Hassan and K.C. Lan, ``Adaptive Position Update in Geographic Routing,&rdquo", *Proc. Int',l Conf. Comm. (ICC ',06)*, pp. 4046-4051, June 2006.

[16] M. Heissenbuttel and T. Braun, "BLR: beacon-less routing algorithm for mobile ad-hoc networks," *Elsevier's Comp. Commun. Journal*, vol. 27, no. 11, pp. 1076-1086, July 2004

[17] S. Ruhrup, "Theory and Practice of Geographic Routing," Department of Computer Science, University of Freiburg, Germany, February 2009

[18] R. Ghosh and J. J. Garcia-Luna-Aceves, "Automatic Routing Using Multiple Prefix Labels," *Proc. IEEE Globecom 2012*, Dec. 3-7, 2012.

[19] S. Funke and N. Milosavljevic, "Guaranteed-Delivery GeographicRouting under Uncertain Node Locations," *Proc. IEEE INFOCOM,2007*

[20] Syue, Syue-Ju; Chin-Liang Wang; Gauthier, V.; Pei-Shan Tsai, "Coverage-Extended Cooperative Geographic Routing with Optimal Power Allocation for Wireless Multihop Networks," *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th* , vol., no., pp.1,6, 6-9 May 2012

[21] Shao Tao; Ananda, A.L.; Chan Mun Choon, "Greedy Face Routing with Face ID Support in Wireless Networks," *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on* , vol., no., pp.625,630, 13-16 Aug. 2007

[22] Ming Li, "Ripple Effect: an Improved Geographic Routing with Local Connectivity Information," *Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian* , vol., no., pp.252,257, 7-10 Dec. 2008

[23] Biao Zhou; Lee, Y.-Z.; Gerla, M., "“Direction” assisted Geographic Routing for mobile ad hoc networks," *Military Communications Conference, 2008. MILCOM 2008. IEEE* , vol., no., pp.1,7, 16-19 Nov. 2008

[24] Rong Ding; Lei Yang, "A Reactive Geographic Routing Protocol for wireless sensor networks," *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on* , vol., no., pp.31,36, 7-10 Dec. 2010

[25] Shengbo Yang; Chai-Kiat Yeo; Bu Sung Lee, "Robust Geographic Routing with Virtual Destination Based Void Handling for MANETs," *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st* , vol., no., pp.1,5, 16-19 May 2010