**Al Albayt University**

**Prince Hussein Bin Abdullah College for Information Technology**

**Department of Computer Science**

# VELOCITY AND CONGESTION-AWARE ROUTING PROTOCOL FOR MOBILE AD-HOC NETWORKS

**بروتوكول لتحديد المسارات في الشبكات الخاصة المتحركة اعتماداً على السرعة والازدحام**

By

**Ahmad Mohammad Al-Maddan**

**Supervisor**

**Prof. Ismail Ababneh**

**This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree of Computer Science**

**Deanship of Graduate Studies**

**Al al-Bayt University**

**First Semester, 2016/2017**

## تفويض

أنا  أحمد محمد عقلة المدّان ،أفوض جامعة آل البيت بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات أو الأشخاص عند طلبهم حسب التعليمات النافذة في الجامعة.

التوقيع :

التاريخ : ٢٠١٧/١/٩م

**إقرار والتزام بقوانين جامعة آل البيت وأنظمتها وتعليماتها لطلبة الماجستير والدكتوراه**

أنا الطالب : أحمد محمد عقله المدّان     الرقم الجامعي : ١٣٢٠٩٠١٠٢٩

التخصص : علم الحاسوب     الكلية : الأمير الحسين بن عبدالله الثاني لتكنولوجيا المعلومات

أعلن بأنني قد التزمت بقوانين جامعة آل البيت وأنظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة بإعداد رسائل الماجستير والدكتوراه عندما قمت شخصيا بإعداد رسالتي بعنوان :

**Velocity and Congestion-aware Routing Protocol For Mobile Ad-hoc Networks**

وذلك بما ينسجم مع الأمانة العلمية المتعارف عليها في كتابة الرسائل والأطاريح العلمية. كما أنني أعلن بأن رسالتي هذه غير منقولة أو مستله من رسائل أو أطاريح أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تحزينها في أية وسيلة إعلامية، وتأسيساً على ما تقدم فأنني أتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في جامعة آل البيت بإلغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني بعد صدورها دون أن يكون لي حق في التظلم او الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

توقيع الطالب :     التاريخ : ٢٠١٧/١/٩م

## COMMITTEE DESCISION

**This Thesis (VELOCITY AND CONGESTION-AWARE ROUTING PROTOCOL FOR MOBILE AD-HOC NETWORKS) was Successfully Defended and Approved on 3/1/2017**

| **Examination Committee** | **Signature** |
|---|---|
| Prof. Ismail Ababneh | ---------------------------- |
| Dr. Saad Bani-Mohammad | ---------------------------- |
| Dr. Omar Shatnawi | ---------------------------- |
| Dr. Wail Mardini | ---------------------------- |

# DEDICATION

I dedicate this thesis to my great father Mohammed Al-Maddan, my lovely mother Sokout, you have successfully made me the person I am becoming.

To my lovely wife Duha Malw-Elain, and my beautiful daughter Mira, you are always the most important part of my success.

To all my family, for their support.

# ACKNOWLEDGMENTS

First and foremost, I would like to thank God for his blessings. I am forever grateful to Prof. Ismail Ababneh for having accepted to be my thesis advisor. His patience, guidance, and assistance were essentials in finishing this work.

Another thank to the examination committee members for their comments that have contributed to improving this work.

# TABLE OF CONTENTS

## Contents

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| Abbreviation | Meaning |
|---|---|
| AODV | Ad hoc On-Demand Distance Vector |
| ABR | Associativity-Based Routing |
| CRP | Congestion adaptive Routing Protocol |
| CARM | Congestion Aware Routing protocol for Mobile ad hoc networks |
| CF | Congestion Factor |
| CMF | Congestion-Movement Factor |
| CBR | Constant Bit Rate |
| DSDV | Destination-Sequence Distance-Vector |
| DLAR | Dynamic Load-Aware Routing |
| DSR | Dynamic Source Routing |
| GPS | Global Positioning System |
| LWR | Load aWare Routing |
| LAR | Location-Aided Routing |
| MANET | Mobile Ad hoc NETwork. |
| OTCL | Object oriented extension Tool Command Language |
| PT | Pause Time |
| RERR | Route Error |
| RREQ | Route Request |
| RREP | Route Reply |
| SSA | Signal Stability-based Adaptive |
| TORA | Temporally-Ordered Routing Algorithm |
| VCAR | Velocity and Congestion-Aware Routing |
| WCD | Weighted Channel Delay |
| WRP | Wireless Routing Protocol |
| ZRP | Zone Routing Protocol |

# LIST OF APPENDICES

# ABSTRACT

Mobile Ad-hoc NETworks (MANETs) are wireless infrastructure-less networks built from mobile nodes. Nodes in MANETs need routes between them when they attempt to communicate. They use a routing algorithm to find and maintain such routes. Most routing protocols proposed for MANETs are designed to find a valid route from source to destination without considering network traffic load and mobility of nodes. There are some algorithms that do consider only congestion when selecting the path from source to destination. The problem with such algorithms is selecting unstable routes, and increases the number of link failures in the network, and consequently increases the packets loss. While there are other algorithms based only on nodes velocity. This type of algorithms selects congested routes for transmitting, and consequently increasing the end-to-end delay of the packets.

In this thesis, we propose a Velocity and Congestion-Aware Routing (VCAR) protocol that aims to select the most stable and least congested intermediate nodes. The VCAR protocol uses two metrics when selecting the best route between source and destination: the congestion level and velocity of intermediate nodes.

To estimate the congestion level at intermediate nodes, we use two metrics: the interface queue length: Qlen, and the number of routes that the node participates in: R. These metrics are measured by the nodes themselves.

We use two approaches to select the route between the source and the destination. The first approach selects the route with the least accumulative value for all intermediate nodes in the path. While the second approach selects the route with the least maximum value among all intermediate nodes in the path.

We implement four versions of VCAR. The first version selects the best route using the least accumulative value of Qlen and velocity for all intermediate nodes. The second version selects the best route using the least maximum value of Qlen and velocity among all intermediate nodes. The third version selects the best route using the least accumulative value of R and velocity for all intermediate nodes, and the fourth version selects the best route using the least maximum value of R and velocity among all intermediate nodes.

The NS2 simulator was used to implement the VCAR protocols, and extensive simulations were conducted to analyze the performance of VCAR variants against the Ad Hoc on-Demand Vector (AODV) protocol. The results show that VCAR variants always outperform AODV in terms of packet delivery ratio, average end-to-end delay, and energy consumption percentage, but the routing overhead was higher.

# CHAPTER ONE: INTRODUCTION.

Wireless communication has been widely used during the last decades, because this type of communication can offer services to users anywhere and at any time. Wireless communication can either have infrastructure or be infrastructure-less. The first class needs an infrastructure, and nodes can only communicate through a central device. While the second one does not need any infrastructure or central administration and is commonly known as ad hoc [15, 23, 26, 28]. MANETs are common infrastructure-less wireless networks [15, 24, 25, 28].

A MANET is usually defined as a set of mobile nodes without any infrastructure or central administration. The nodes are free to move around in any direction. Each node in a MANET has a limited transmission range and can operate as a host and as a router [8, 13, 14, 15, 28]. The node can directly send a message to all other nodes within its transmission range, but for the nodes outside the transmission range, the message should be forwarded to the destination via other intermediate nodes in the MANET [8, 23]. To route a messages in the MANET, a routing algorithm is needed to allow the node to discover and maintain a route to the destination.

## 1-1MANET Characteristics

MANETs have several characteristics that are very important to be understood when studying the nature of this type of networks [15, 28]. The following summarizes these characteristics:

1. Each node in a MANET has a limited transmission range [4]. A route is needed to reach nodes outside this range. As shown in Figure 1-1, node B lies within the transmission range of node A, so a direct message can be sent from node A to node B,

while a route is needed between node A and node C because node C lies outside the transmission range of node A.

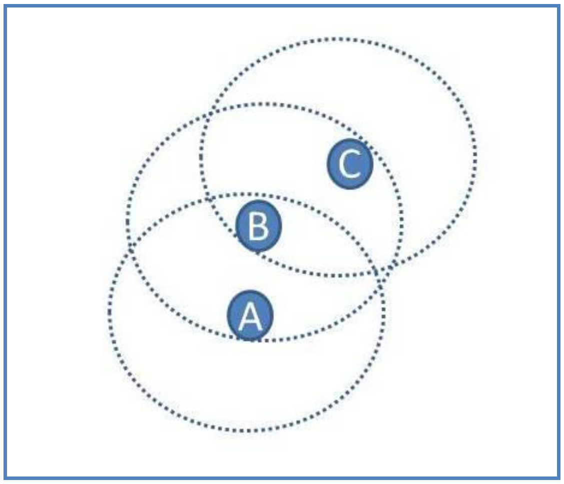


**Figure 1-1: Node Transmission Range**

2. Each node in a MANET is battery-powered and the power is quickly consumed because each node acts as a host and as a router [4].

3. The nodes in a MANET are free to move around in any direction and this makes the topology change frequently. This characteristic should be taken into account when designing the routing algorithm. Figure 1-2 shows the effect of mobility on the communication of nodes. In Figure 1-2(b) node B has moved so that it is now outside the transmission range of node A. When node A needs to communicate with node B, it must send the message to node C, which sends it to node B.

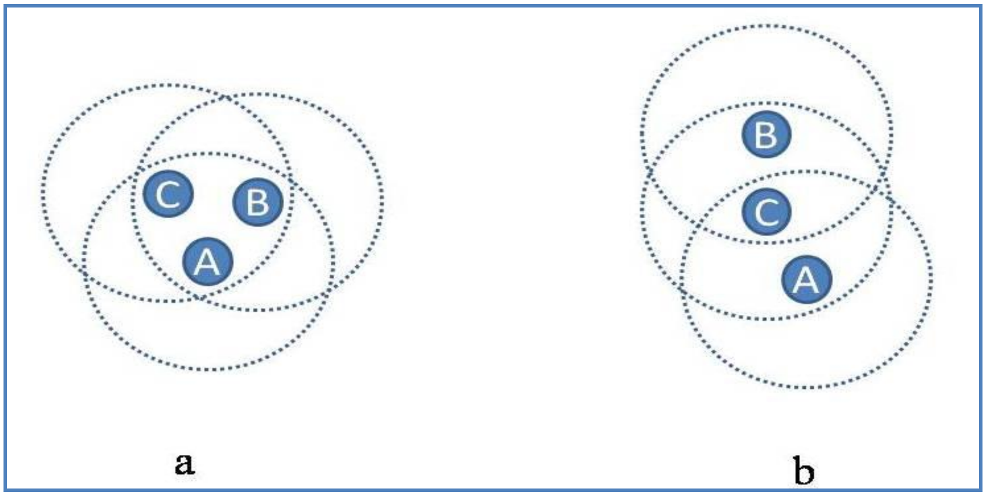


**Figure 1-2: Node Mobility Causes Topology Change in MANET**

## 1-2MANET Applications

MANETs can be used when building a fixed network is expensive or hard to do [25]. MANETs have been proposed for use in many industrial and commercial fields [24]:

1. MANETs have been used in wireless sensor network applications. These applications are used mainly in environmental fields, such as data tracking and remote sensing for weather forecasting.

2. MANETs have been proposed for some civilian applications, such as search and rescue operations, and disaster relief efforts in areas where there is no network infrastructure.

3. MANETs have been used in military applications because of their rapid deployment capability.

## 1-3 Routing in MANETs

When studying MANETs, one of the most important questions is how mobile nodes can communicate with each other without any infra-structure using wireless media. The answer is that each node needs to maintain a routing path to nodes with which it needs to communicate. Many routing algorithms were proposed for MANETs.

One of the most important issues that must be considered when designing any routing algorithm is selecting a route that can adapt well to topology changes [13, 29].

Routing influences the overall performance of the MANET. A more efficient routing algorithm leads to a better MANET performance. Routing in MANETs should take into account the mobility of the nodes because nodes are free to move around. The routing information should be updated dynamically. There are mainly three types of routing protocols in MANETs: proactive, reactive, and hybrid [5, 8, 10, 15, 23, 28, 30].

### 1-3-1 Proactive Routing Protocols.

In proactive routing, each node stores routing information in a table, so this type is also called table-driven routing [5, 23, 30]. When a node needs a route to another node, it finds the route by searching in its routing table. The routing table information is propagated periodically through the network, so that each node has an updated routing table. This type of routing has low latency because the routes are always available in the table, but it suffers high overhead because of periodic routing table updates. Examples of such type of routing include Destination-Sequence Distance-Vector Routing (DSDV) [20], and Wireless Routing Protocol (WRP) [17].

## 1-3-2 Reactive Routing Protocols.

In reactive routing, the route is determined only when needed, so that it is also called source-initiated on-demand routing [5, 23, 30]. When a node needs to send a message to another node for which it does not have a known route, it first finds a route to that node by flooding the network with a Route Request (RREQ) packet, and then it uses the discovered route to send the message. This type of routing has high latency. However, it has low overhead because routes are determined only when needed. Examples of routing protocols of this type include Ad hoc On-Demand Distance Vector (AODV) protocol [21], and Dynamic Source Routing (DSR) protocol [9].

## 1-3-3 Hybrid Routing Protocols.

This type of routing combines the advantages of the above two types. In hybrid routing, the routes between nodes are initially discovered proactively, then they are maintained reactively as a result of link failures or topology changes [5, 23, 30]. Examples of this type of routing protocols include Zone Routing Protocol (ZRP) [27], and Temporally-Ordered Routing Algorithm (TORA) [19].

## 1-3-4 Ad-hoc On-demand Distance Vector (AODV) Routing Protocol.

AODV is a reactive routing protocol proposed in [21]. AODV uses a route discovery process to find the shortest path between the source and the destination. AODV consists of the following phases.

- **Route Discovery Phase**.

When the source does not know a valid route to the destination it broadcasts a route request (RREQ) packets to all of its neighbors. The RREQ packet has a unique sequence number to detect duplicate packets. It also includes some other information such as: destination identifier, source identifier, and time to live. Each intermediate node that has been received the RREQ uses the sequence number field to determine if the received RREQ has been previously received or not. Then it will check if it has a valid route to the destination, if so it will send a Route Reply (RREP) packet to the source, otherwise it will rebroadcasts the RREQ packet to all of its neighbors, and the time to live field will decremented by one to prevent packet looping. The RREQ packet continue traveling until it reaches to the destination. When the destination receives the RREQ, it replies with a Route Reply Route (RREP) packet to the intermediate node from which it receives the RREQ. The RREP travels back until it received by the source. [21].

- **Route Maintenance Phase.**

The route maintenance phase is responsible for detecting any link failure. The node detects any link failure by listening to hello messages from its neighbors. If any link failure is detected, a node broadcasts a Route Erorr (RERR) packet to notify the source that it detects a link break, so that the source will initiate a new RREQ packet to search for a valid route to the destination [25].

## 1-4 the Problem and Motivation.

Most of reactive protocols proposed for MANETs have been designed to choose the shortest path from source to destination [14]. The shortest path is the path with the smallest number of hops between the source and destination. Using this single metric can lead to a situation in which the nodes that participate in the shortest path are congested while other nodes are idle or lightly-loaded [14, 16]. Figure 1-3 shows such situation in which node D lies in the shortest path in most of the routes because of its location in the middle section of the MANET, and this increases the probability of congestion at node D. For example, the shortest path from A to F is A->D->F, while other routes are available such as A->C->E->F and A->B->G->F. To avoid such problems, other metrics should be used when selecting the best path.

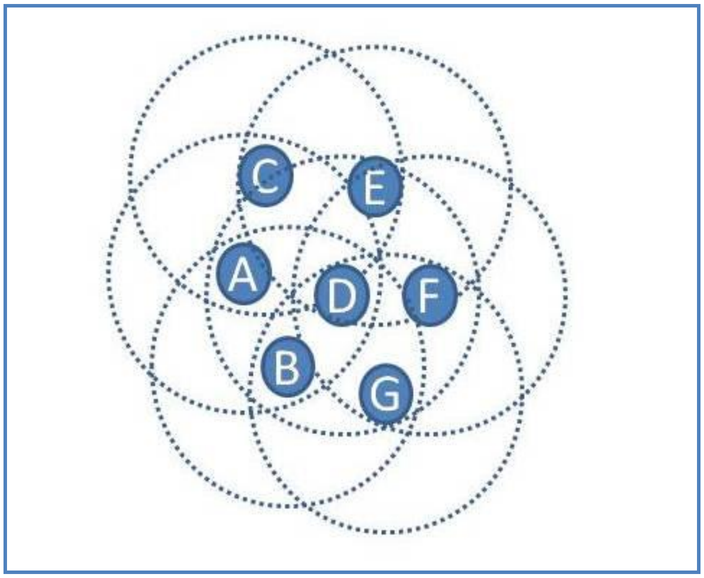


**Figure 1-3: Shortest Path Leads to Congested Nodes**

Another problem with such protocols is that nodes in MANET may often move around. The movement of nodes results in dynamic changes in routes, as shown previously in Figure 1-2, and this requires finding new routes. Continuous unpredictable changes in MANET topology increase the overhead in the route maintenance mechanism [13, 29].

To increase the performance of the routing protocol, the route discovery mechanism should take into account both the congestion state and movement of the nodes. The routes in a MANET should be as stable as possible. The metric used in route selection should combine both the congestion level and the movement of nodes in the path selected. Any node participating in a route should have low congestion and low speed.

## 1-5 The Objectives of The Study

The purpose of this study is to propose a new routing protocol for MANETs called Velocity and Congestion-Aware Routing (VCAR) protocol. VCAR selects the routes between nodes according to both the congestion and velocity of intermediate nodes. VCAR aims to achieve the following goals:

- Reduce the congestion in the network.
- Increase the packet delivery ratio in the network.
- Decrease the average end-to-end delay of packets.
- Decrease the energy consumption of the nodes.

# CHAPTER TWO: RELATED WORKS

Routes determined by reactive routing protocols can be congested. Many algorithms have been proposed to solve this congestion problem Most of these algorithms use some metric to measure node congestion, and select the least congested node or route [8, 16, 26].

Routes can be unstable because of node movement in MANETs. Therefore, some routing algorithms use the mobility metric to enhance network performance. These algorithms try to choose route nodes with low mobility so as to discover relatively stable paths. A more stable path decreases the path failure rate and increases the throughput of the network [5].

## 2-1 Associativity-Based Routing Protocol.

The Associativity-Based Routing (ABR) [29] protocol uses node stability as the main metric in selecting the best path. ABR is a reactive protocol that searches for a route only when a source needs to find a route to a destination. ABR uses an associativity-based scheme in which a route is constructed from nodes that have an associativity state with their neighbors that guarantees stability. Thus, ABR selects routes that are likely to be long-lived. ABR measures node stability by the node's association with its neighbors. ABR finds all the possible routes from source to destination, and then selects the best path according to the selection criteria. The disadvantage of ABR is that it does not consider node congestion in the route selecting process; this may lead to using a route with congested nodes.

## 2-2 Signal Stability-Based Adaptive Routing Protocol.

The Signal Stability-Based Adaptive (SSA) [3] routing protocol is another on-demand protocol that proposed to perform route discovery by selecting the longest-lived path between source and destination. SSA selects the best route by including signal strength and node location stability. This protocol ranks the channels as strong and weak according to their average signal strength between the two ends of the channel. The node location stability is used by SSA so as to choose the longest-lived route. By considering these two criteria, SSA always chooses strong channels that have existed for a period of time that is greater than some threshold value identified by the protocol. In SSA, the source broadcasts the RREQ packet to all of its neighbors. When any intermediate node receives the request packet, it rebroadcasts it only if it was received over a strong channel. The destination chooses the route of the first arriving request because it is probably shorter and less congested, and sends a reply message that contains the best route. The main disadvantage of SSA is that it works fine only when there are a significant number of strong routes between source and destination.

## 2-3 Location-Aided Routing Protocol.

The Location-Aided Routing (LAR) [13] protocol was proposed to improve the performance of routing discovery by utilizing the location information of the nodes that are obtained using a Global Positioning System (GPS). LAR uses location information to limit the search of a new route to a small area in the MANET. This small area is called request zone. LAR aims to reduce the overhead by decreasing the number of control packets in the discovery process. Once the request zone is identified, the source searches only within this zone, without flooding all the MANET with RREQ packets. The request zone is rectangular in shape and is identified by the source. When the source wants to discover a route to the destination, it must identify the request zone of the destination by using some information that includes the previous location and the average speed of the destination. After identifying the request zone, the source broadcasts the RREQ packet only to the nodes that are located within the request zone boundaries. LAR performance depends on both the availability and accuracy of GPS used, and the need to use GPS is the major disadvantage of LAR.

## 2-4 Dynamic Load-Aware Routing protocol.

Lee and Gerla proposed the Dynamic Load-Aware Routing (DLAR) [14] protocol in 2001. DLAR is a reactive protocol that uses the routing loads of nodes as the main metric to select the best path. In this protocol, the highly loaded nodes are prevented from participating in the routes. The load of a node is measured by the number of packets buffered in its interface queue. This protocol selects always the more stable path, and reduces end-to-end delay. DLAR also controls the congestion states of the active routes and reconstructs them when any node in the path reaches its maximum queue capacity. In DLAR, the source broadcasts a request packet to find a route to the destination, and each intermediate node appends its load and broadcasts the packet again. The receiver can select the best path according to the loads of the intermediate nodes. It selects the path with the lowest aggregate load. The main disadvantage of DLAR is that it has a large overhead caused by the request flooding process in which each intermediate node appends extra information before rebroadcasting the request.

## 2-5 Load aWare Routing Protocol.

In 2003, a new protocol was proposed by Yi and Gerla. It is called the Load aWare Routing (LWR) [31] protocol. LWR was built using the same idea as DLAR and was proposed to solve the problem of large overhead. LWR is different from DLAR in that each intermediate node has a routing selection scheme with a congestion control mechanism at the same time. In LWR, intermediate nodes drop any request packet when they are overloaded. Another difference is that LWR concentrates on the route discovery phase to prevent any unnecessary broadcasting of requests, while DLAR does this in the route reply phase. LWR decreases the overhead and improves the routing performance.

## 2-6 Congestion Adaptive Routing Protocol.

The Congestion adaptive Routing Protocol (CRP) [30] is an adaptive routing protocol that tries to avoid congestion rather than dealing with it reactively. The main idea in CRP is that each node in any route should notify its previous node when it is about to be congested. To prevent congestion, CRP introduces a secondary route called the bypass route. CRP uses the bypass route to bypass the congestion area to the first non-congested node on the primary route. The traffic is split over these two routes. CRP has a significant overhead when there is a bypass route for each primary one. Another disadvantage of CRP is that there is a small loss rate because of dividing the traffic into two routes.

## 2-7 Congestion Aware Routing Protocol.

The Congestion Aware Routing protocol for Mobile ad hoc networks (CARM) [2] is another congestion-aware protocol that introduces a new parameter to measure the congestion level. This parameter is called the Weighted Channel Delay (WCD). WCD is calculated using two weighted parameters: the total time spent at the MAC layer and the number of buffered packets in the node, as shown in Equation 2-1.

$$WCD = aQ + bT \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \text{Equation 2-1}$$

Where  is the number of buffered packets, $T$ is the total time spent at the MAC

layer, and $a$ and  are constants between 0 and 1.

In CARM, the source node broadcasts a RREQ packet. Each intermediate node calculates its WCD, appends it in the request packet, and rebroadcasts the packet. When the destination receives the request, it responds with a reply message that contains the best route by choosing the smallest aggregate value of WCD of all intermediate nodes. CARM improves the adaptability to congestion by using the WCD parameter, which is used for selecting routes with high throughput and low congestion. On the other hand, CARM causes overhead at intermediate nodes.

## 2-8 Congestion-Aware Routing Protocols for Ad Hoc networks.

Another congestion-aware routing protocol was proposed by Seetan, Ibabneh, and Dala'ah in their paper [25] called Min_Total_CA protocol. In this protocol the congestion is measured using the number of routes that the node participates in. Min_Total_CA selects the route with the least total value of congestion among all available routes between the source and the destination. Each route has two congestion value: the forward congestion value, and the backward congestion value. The first value is calculated through the route discovery process, and the second value is calculated through the route reply process. When the source does not have a valid route to the destination, it broadcasts a RREQ packet to all of its neighbors. Each intermediate node receives a RREQ packet, will reply with a RREP packet if it has a valid route to the destination. Otherwise, it adds its forward congestion value to the RREQ packet. The RREQ packet travels through the network until it reaches to the destination. The destination checks if it has a valid route to the source, if not it replies with a RREP packet. But if there is a valid route, then the destination will send RREP packet only if the received forward congestion value is smaller than that in the currently used route. The RREP travels back to the source, and each node in the reverse path adds its backward congestion value to the backward congestion value in the RREP. When the source receives a RREP packet from the destination, it checks if it's the first RREP, if so it starts using this route. Otherwise, it checks the freshness of the RREP by checking if the new reply has a destination sequence number that larger than that in the route currently in use, if so it starts using the new route. If the new and the current routes have the same freshness, the source checks if the backward congestion value of the received RREP is lower than that in the current route, if so it starts using the new route.

# CHAPTER THREE: THE PROPOSED STUDY

## 3-1 Velocity and Congestion-Aware Routing Protocol

In this section, we present the design of VCAR protocol and illustrate the network model, congestion metrics, and node movement metrics.

### 3-1-1 Network model.

We consider a MANET that contains mobile nodes with bidirectional wireless links connecting them. All nodes have the same transmission range, initial battery life, processing capacity, mobility pattern, transmission bandwidth, and responsibilities (each node can route packets). Each node has its own movement speed, and movement direction.

### 3-1-2 Congestion Metrics.

To address the congestion level at each node, we propose to use two metrics: Interface Queue Length (Qlen), and number of Routes (R) that the node participates in. These two metrics can be measured by the node itself so as to measure the congestion level.

- Queue Length (Qlen).

Each node has an interface that is used to communicate with other nodes. Packets are queued in the interface until they have been transmitted. At any given time, there are a number of waiting packets in the queue (Qlen). Lower Qlen value indicates lower congestion level, and vice versa.

- Number of Routes (R).

All nodes can act as routers, so any node could participate in a number of routes. To measure the congestion level at a node, it is important to count how many routes each node participates in. A lower value of R means lower node congestion level, and vice versa.

VCAR uses either Qlen or R to measure the congestion level at each node. One of our contributions is to evaluate the use of these factors along with the node movement factor (which will be illustrated in the following subsection). For the rest of this thesis we will use the term Congestion Factor (CF) to indicate either Qlen or R.

### 3-1-3 Node Movement Metric.

In MANET routes, nodes that do not move frequently will provide better performance than other nodes because they reduce the overhead in construction and reconstruction of routes. The routes that are composed of slow nodes are more stable than other routes. Our proposed algorithm attempts to use slow nodes in constructing routes between nodes. We use node Velocity (V) as an indicator of node stability level. A lower value of V indicates higher node stability, and vice versa.

### 3-1-4 VCAR Design

VCAR reactively discovers routes from source to destination and selects the best route according to the values of CF and V. The main metric for selecting the best route is called the Congestion-Movement Factor (CMF). CMF is calculated as per Equation 3-1.

$$\mathbf{CMF = \alpha * CF + (1 - \alpha) * \frac{V}{Vmax}} \qquad \text{Equation 3-1.}$$

Where, *Vmax* is the maximum velocity of a node.

The value of $(\alpha)$ is between 0 and 1. This value should be tuned to achieve good performance.

There are two approaches to select the best route between the source and destination. The first approach selects the route with the least accumulative CMF value for all nodes in the path. While the second approach selects the route with the least maximum CMF value.

To differentiate between these two approaches, suppose we have the MANET shown in Figure 3-1. Each node is associated with its CMF value.

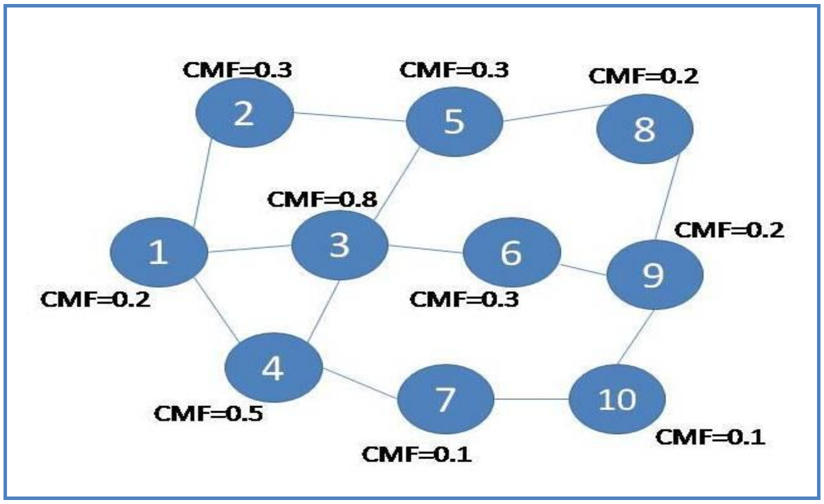


**Figure 3-1: Each Node in a MANET Has its Own CMF Value**

Suppose that node 1 needs to send data to node 9 and it has not a valid route to that node. Then it should discover a new route. There are many possible routes from node 1 to node 9. Each one of these possible routes has an accumulative CMF value for all intermediate nodes, and a maximum CMF among all the intermediate nodes, as shown in Table 3-1.

**Table 3-1: Accumulative and Maximum CMF Values**

| Route | Accumulative CMF | Maximum CMF |
|---|---|---|
| 1,2,5,8,9 | 0.8 | 0.3 |

| | | |
|---|---|---|
| 1,2,5,3,6,9 | 1.7 | 0.8 |
| 1,2,5,3,4,7,10,9 | 2.1 | 0.8 |
| 1,3,5,8,9 | 1.3 | 0.8 |
| 1,3,6,9 | 1.1 | 0.8 |
| 1,3,4,7,10,9 | 1.5 | 0.8 |
| 1,4,3,5,8,9 | 1.8 | 0.8 |
| 1,4,7,10,9 | 0.7 | 0.5 |
| 1,4,3,6,9 | 1.6 | 0.8 |

If we use the least accumulative CMF approach, then the best route will be 1,4,7,10,9 because it has the least accumulative CMF value which is (0.7). On the other hand, If we use the least maximum CMF approach, then the best route will be 1,2,5,8,9 because it has the least maximum CMF value, which is (0.3).

VCAR consists of three phases: the route discovery phase, the route reply phase, and the route maintenance phase. In the first phase, the sender broadcasts a RREQ packet to discover a route to the designated destination. In the second phase, a Route Reply (RREP) packet is sent to the source to identify the selected route to the destination. Finally, the third one is responsible for sending a Route Error (RERR) packet when a route becomes invalid for any reason. These phases will be illustrated in details later in this section, but before that we will present some important information about routing control packets in the following sub-section.

### 3-1-4-1 VCAR Control Packets

There are three types of control packets: RREQ, RREP, and RERR. In this subsection, we will presents the main fields of each one.

The RREQ packet contains the following main fields:

- Source Identification (SID): the address of the source node from which the route need to be established.
- Destination Identification (DID): the address of the destination node, to which a packet or more are to be sent.
- Sequence Number (SEQ): a number that uniquely identifies each packet sent from a source. This number is used to detect duplicate RREQ packets.
- CMF: each node that receives a RREQ calculates its CMF as in Equation 3-1, and then updates the CMF field. The CMF field contains either the cumulative CMF values for all the nodes visited by the RREQ or the maximum CMF value among the CMF values of all the nodes visited by the RREQ.
- Number of Hops (HOP): number of hops traversed by the RREQ.
- Time To Live (TTL): represents the maximum number of hops that the RREQ can traverse before its lifetime expires. This value is used to prevent looping of packets. It is decremented by one at each node receiving the RREQ.

The RREP packet contains the following main fields:

- Source Identification (SID): the address of the destination of the RREQ.
- Destination Identification (DID): the address of the source node that initiated the RREQ.
- Sequence Number (SEQ): is a number that is uniquely assigned to a packet. This number is used to detect duplicate RREP packets.
- CMF: contains either the cumulative or the maximum CMF, as explained earlier.
- Number of Hops (HOP): number of hops traversed by the RREQ.

The RERR packet is the same as in AODV protocol [21].

### 3-1-4-2 Route Discovery Phase.

The route discovery phase is responsible for finding a route between two nodes: source and destination. The following steps describe this phase.

**1.** If the source node does not have a valid route to the destination, it prepares a RREQ packet and broadcasts it to all neighbor nodes.

**2.** If an intermediate node receives a RREQ packet, then it does one of the following:

**2.1** If TTL in RREQ is 0 or SEQ in RREQ had been processed previously, the node discards the RREQ.

**2.2** If the DID in RREQ is equal to the intermediate node's ID (i.e., the intermediate node is the destination node), the node replies with a RREP packet as will be explained in the route reply phase.

**2.3** If the intermediate node is not the destination node, it does one of the following:

**2.3.1** if the node has a valid route to the destination, the node replies with a RREP packet as will be explained in the route reply phase.

**2.3.2** if the node does not have a valid route to the destination, the node takes the following actions:

**2.3.2.A** It calculates its CMF as expressed in Equation 3-1.

**2.3.2.B** It updates the value of CMF in the RREQ.

**2.3.2.C** The node rebroadcasts RREQ to all of its neighbors.

### 3-1-4-3 Route Reply Phase.

This phase is entered when an intermediate node has a valid route to the destination or the RREQ has arrived to the destination. The following steps describe this phase.

**1.** The responding node does the following:

**1.1** It calculates CMF as expressed in equation 3-1.

**1.2** It calculates the CMF value of the path, either accumulative or the maximum.

**1.3** If it's the first RREP sent to the source, then go to step 1.5.

**1.4** Otherwise, the node compares the calculated CMF with the last sent CMF value. If the current one is larger than the last one, then discard the packet. Else, go to step 1.5.

**1.5** Node prepares a RREP packet that contains the CMF value.

**1.6** The node sends its RREP to the node from which it received the RREQ.

**2.** The RREP message travels back to the source node that initiated the RREQ via all the nodes that previously rebroadcasted the RREQ packet.

### 3-1-4-4 Route Maintenance Phase.

The Route maintenance phase is responsible for detecting any link failure. Usually, link failure happens due to node movement. This phase is done as in AODV [21].

### 3-1-4-5 VCAR Versions

As we mentioned in section 3-3-2, we have two congestion metrics: queue length, and number of routes, and there are two approaches to select the best path: the least accumulative CMF value, and the least maximum CMF value. So that VCAR has four versions. Each version uses a congestion metric along with a measurement approach as shown in Table 3-2. We will use the names in the table to distinguish each one of the four protocols.

**Table 3-2: VCAR Versions**

| Version | Congestion metric | Best Path Selection Approach | Protocol Name |
|---|---|---|---|
| 1 | queue length | least accumulative CMF | VCAR_Q_ACC |
| 2 | queue length | least maximum CMF | VCAR_Q_MAX |
| 3 | number of routes | least accumulative CMF | VCAR_R_ACC |
| 4 | number of routes | least maximum CMF | VCAR_R_MAX |

# CHAPTER FOUR: THE SIMULATION

Network simulators are used mainly to compare the performance of different routing protocols used in MANETs. NS-2, OPNET, QualNet, and GloMoSim are the most popular simulators used in this field. We used the NS-2 simulator to evaluate the performance of VCAR protocols against AODV. NS-2 is an accurate, event driven, and open source simulator. It contains the implantation of many existing routing protocols such as DSR and AODV [6].

## 4-1 NS-2 simulator Architecture

NS-2 is an object-oriented simulator, written in the C++ programming language, with an Object oriented extension Tool Command Language (OTCL) interpreter [18]. C++ is used mainly in the implementation of network protocols. While OTCL is used to write the simulation scenarios [6, 18].

The results from any simulation scenario are stored in a text files called trace files. These files contains all the events that take place in the network during the simulation. Another tool, called GAWK, was used to extract the needed information from these trace files.

## 4-2 Simulation environment

The NS-2 simulator uses a model for each layer [6]. We have used the models shown in Table 4-1.

**Table 4-1: Models Used for Different Layers**

| Layer | Model |
|---|---|
| Application | CBR |
| Transport | UDP |
| Mac Layer | 802.11 |

The simulation parameters include the physical channel specifications, mobility models, and network traffic. All simulations were performed using NS-2.35 version. Table 4-2 shows the simulation parameters used in the study, this environment is commonly used, and recommended in many studies such as [14, 25].

**Table 4-2: The simulation parameters**

| Parameter | Value |
|---|---|
| Simulation time | 300 seconds |
| Number of mobile nodes | 50 nodes |
| Simulation area | 1200 meters * 1200 meters |
| Node transmission range | 250 meters |
| Maximum buffer queue | 50 packets |
| Maximum speed of nodes | 10 meters/second |
| Data flow type | CBR |
| Number of sending sources | 5, 10, and 15 sources |
| Sending rate | 1, 2, 4, and 6 packets/second |
| Routing protocol | AODV, VCAR_Q_ACC, VCAR_Q_MAX, |

| | |
|---|---|
| | VCAR_R_ACC, and VCAR_R_MAX |
| Weight factor used in VCAR protocols (α) | 0.25, 0.50, and 0.75 |

The following explains each one of these parameters:

1. Simulation time: specifies the total simulation time in seconds. All the simulations lasted for 300 seconds.
2. Simulation area: specifies the dimensions of the simulation area. We used simulation area of 1200 meters * 1200 meters.
3. Number of nodes: specifies how many nodes were in the simulation area. We ran all the simulations with 50 nodes.
4. Mobility model: specifies the style of node mobility. NS-2 supports different mobility styles. The most widely used mobility style is the random-waypoint style. In this type of mobility, a node randomly chooses a destination in the simulation area, and moves toward this destination with a random velocity uniformly chosen between two values, generated as discussed below. When the node reaches its destination, it stays there for a period of time called Pause Time (PT). Then, it selects another destination and moves toward it. In this research we used random-waypoint style with PS values of: 0,100, 200, and 300 seconds. Node velocity is generated uniformly between 0 and 10 meters/second.
5. Traffic model: specifies the communication model used for transmitting packets. We used the Constant Bit Rate (CBR) model for sending packets from the source to the destination. Each packet is 512 bytes. In the CBR model a certain number of nodes

called communication sources send packets to other nodes at some rate. In this study, we use 5, 10, and 15 sources, with sending rates of 1, 2, 4, and 6 packets/second.

6. Transport layer protocol: specifies the transport protocol used in the simulation. We used UDP to ensure timely delivery of data packets with low network overhead.
7. Routing protocol: specifies the routing protocols used in the simulation. We ran the simulations using AODV, and VCAR protocols.

## 4-3 Experiments Design

For each experiment, we create two files: traffic connection file, and node movement file. In the following two subsections, we illustrate how these two files were created.

### 4-3-1 Creating Traffic Connection File

The traffic connection file contains a number of CBR traffic connections generated using the command "ns cbrgen.tcl". The general format of this command is as follows [6]:

*ns cbrgen.tcl. [-type cbr] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]*

Where *type* specifies the traffic type, *nn* specifies the number of nodes in the simulation, *mc* specifies the number of traffic connections, and *rat*e specifies the sending rate (packets/second).

As example, the following command creates 5 CBR traffic connections, each connection sends 1 packet/second.

*ns cbrgen.tcl. -type cbr -nn 50 -seed 1 -mc 5 -rate 1*

We created twelve traffic connection files to accommodate all the cases in the study. Each file was created using a combination of a number of communication sources and a sending rate as shown in Table 4-3.

**Table 4-3: The Traffic Connection Files**

| File name | Number of communication sources | Sending rate (packets/second) |
|---|---|---|
| cbr-mc_5-rate_1 | 5 | 1 |
| cbr-mc_5-rate_2 | 5 | 2 |
| cbr-mc_5-rate_4 | 5 | 4 |
| cbr-mc_5-rate_6 | 5 | 6 |
| cbr-mc_10-rate_1 | 10 | 1 |
| cbr-mc_10-rate_2 | 10 | 2 |
| cbr-mc_10-rate_4 | 10 | 4 |
| cbr-mc_10-rate_6 | 10 | 6 |
| cbr-mc_15-rate_1 | 15 | 1 |
| cbr-mc_15-rate_2 | 15 | 2 |
| cbr-mc_15-rate_4 | 15 | 4 |
| cbr-mc_15-rate_6 | 15 | 6 |

## 4-3-2 Creating nodes movement file

The nodes movement file contains the movement of nodes within the simulation area according to the random-waypoint mobility style. This file is generated using the command *"setdest"*. The general format of this command is as follows [6]:

*setdest [-n num_of_nodes] [-p pausetime] [-m maxspeed] [-t simtime] [-x maxx] [-y maxy]*

Where *n* specifies the number of nodes in the simulation, *p* specifies the pause time, *m* specifies the maximum speed, *t* specifies the simulation time, and *x* and *y* specify the dimension of the simulation area.

As example, the following command creates the node movement file for 50 nodes that move in an area of 500 meters* 500 meters, and follow the random waypoint style with a pause time=0 and a velocity generated randomly between 0 and 10 meters/second. The simulation lasts for 300 seconds.

*setdest -n 50 -p 0.0 -m 10.0 -t 300 -x 500 -y 500*

Because we have four pause times in the study: 0, 100, 200, and 300 seconds, then we create four corresponding files as shown in Table 4-4.

**Table 4-4: Nodes Movement Files**

| Movement file | Pause time (seconds) |
|---|---|
| scen-0 | 0 |
| scen-100 | 100 |
| scen-200 | 200 |
| scen-300 | 300 |

To cover all the cases, we create forty eight simulation scenarios. Each scenario exports one traffic file and one movement file.

## 4-4 Performance Comparison Metrics

Gawk tool was used to extract data from the trace files generated by the simulations. The performance analysis uses four metrics: packet delivery ratio, average end-to-end delay, routing overhead, and energy consumption percentage.

### 4-4-1 Packet Delivery Ratio

The packet delivery ratio is the ratio between the total number of received data packets to the total number of sent data packets [1, 7, 10, 11, 12]. As per equation 4-1.

…. Equation 4-1

### 4-4-2 Average End-to-End Delay

End-to-end delay is the delay that a packet suffers between the time it leaves the source application to the time it arrives at the destination application. The average end-to-end delay is the average value of such delays suffered by all the data packets received in the network [1, 7, 10, 11, 12]. It is computed as per Equation 4-2.

$$\textbf{Average End} - \textbf{to} - \textbf{End Delay} = \frac{\sum_{i=1}^{P}(RTi - STi)}{P}$$ …………………. Equation 4-2

Where, *P* is the total number of data packets received in the network, *RTi* is the time at which packet *Pi* was received, and *STi* is the time at which packet *Pi* was sent.

## 4-4-3 Routing overhead

Routing overhead is the ratio between the total number of control packets sent to the total number of data packets received [1, 7, 10, 11, 12]. It is computed as per Equation 4-3.

$$\text{Routing Overhead} = \frac{\text{total number of control packets sent.}}{\text{total number of datapackets received.}}$$ ……………………Equation 4-3

## 4-4-4 Energy Consumption Percentage

Energy consumption percentage is the ratio of the total energy consumed at all the nodes to the total initial energy of all the nodes. Energy consumption for each node is defined as the difference between its initial energy and the remaining energy at the end of the simulation. It is computed as per Equation 4-4.

$$\text{Eneregy Consumption Percantage} = \frac{\sum_{i=1}^{N}(IE - REi)}{N * IE} * 100\%$$ ……………………Equation 4-4

Where *N* is the number of nodes in the network, *IE* is the initial energy of the node which is the same for all nodes, and *REi* is the remaining energy at node *i*.

# CHAPTER FIVE: RESULTS AND DISCUSSION

In this chapter, we present and analyze the results obtained from the simulation scenarios and evaluate the performance of VCAR against other protocols. We ran the simulations using three values of (α): 0.25, 0.50, and 0.75. Empirically, we have found that α=0.50 gives the best overall results. The following sections present and discuss the results obtained from the simulation scenarios when α=0.50. To see the results when α=0.25, and α=0.75, please refer to appendix A, and appendix B, respectively. Each experiment was ran ten times, and the average value of these runs was considered.

We evaluate the performance of VCAR against a previous congestion-aware routing protocol called Min_Total_CA that was proposed by Seetan, Ababaneh, and Dalal'ah in [25]. We used the simulation environment that illustrated previously in section 4-2, and considered the average value for all the simulation scenarios. The results show that VCAR has achieved a better packet delivery ratio than Min_Total_CA. It has improved the packet delivery ratio by 6.50 percent. VCAR also outperforms Min_Total_CA by 6.00 percent in terms of average end-to-end delay. The results show that VCAR has a lower energy consumption than Min_Total_CA, with an improvement of 8.25 percent. The results also show that the routing overhead values for both VCAR and Min_Total_CA are fairly close to each other.

In the following sections we discuss the results of the simulations and evaluate VCAR performance against AODV performance.

## 5-1 Packet Delivery Ratio

Figures 5-1 ~ 5-12 show the packet delivery ratio for different transmission rate values and a different number of sources. The results show that all VCAR protocols outperform AODV in terms of packet delivery ratio for all mobility levels, all number of sources, and all transmission rates because they select less congested and more stable routes among all available routes, which reduces the number of dropped packets.

We noticed that all VCAR protocols outperform AODV more substantially when mobility is high and the load is heavy. On the other hand, they outperform AODV less when the mobility is low and the load is light. As example, when the number of sources is fifteen and each source sends six packets per second, VCAR_R_ACC outperforms AODV by 26.99 percent. On the other hand, when the number of sources is five and each source sends one packet per second, VCAR_R_ACC outperforms AODV by 13.94 percent.

The VCAR_R_ACC and VCAR_R_MAX protocols always have a better packet delivery ratio than VCAR_Q_ACC and VCAR_Q_MAX. The reason is that the last two versions (VCAR_Q_ACC and VCAR_Q_MAX) use buffer length to measure the congestion level at the intermediate nodes, and because the buffers at intermediate nodes are almost full (especially when the traffic is heavy) then these two protocols could not select the least congested route accurately.

The VCAR_R_ACC protocol always outperforms all other VCAR protocols in all the cases we have studied. On average, VCAR_R_ACC outperforms AODV by 15.63 percent for low mobility networks, and by 21.63 percent for high mobility networks.

Figure 5-1 shows the packet delivery ratio for all protocols when the number of sources is five and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 13.94, 9.27, 4.89, and 1.84 percent, respectively. When pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 4.6, 4.44, 2.95, and .61 percent, respectively.

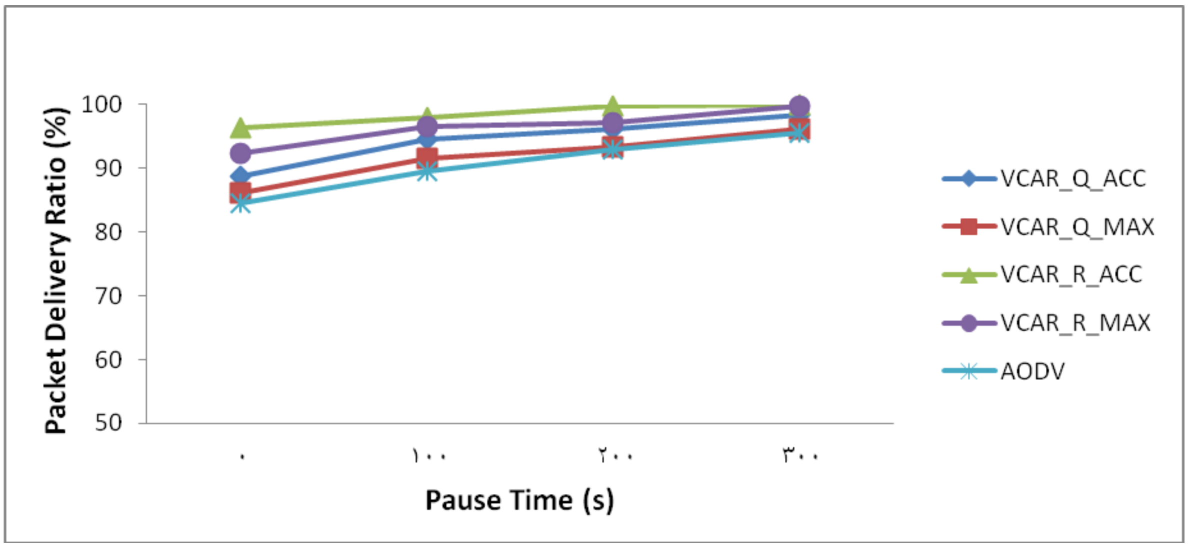


**Figure 5-1: Packet Delivery Ratio of 5 sources each one sends 1 packet/s**

Figure 5-2 shows the packet delivery ratio for all protocols when the number of sources is five and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 15.14, 10.43, 5.99, and 2.91 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 10.67, 8.33, 6.78, and 4.35 percent, respectively.

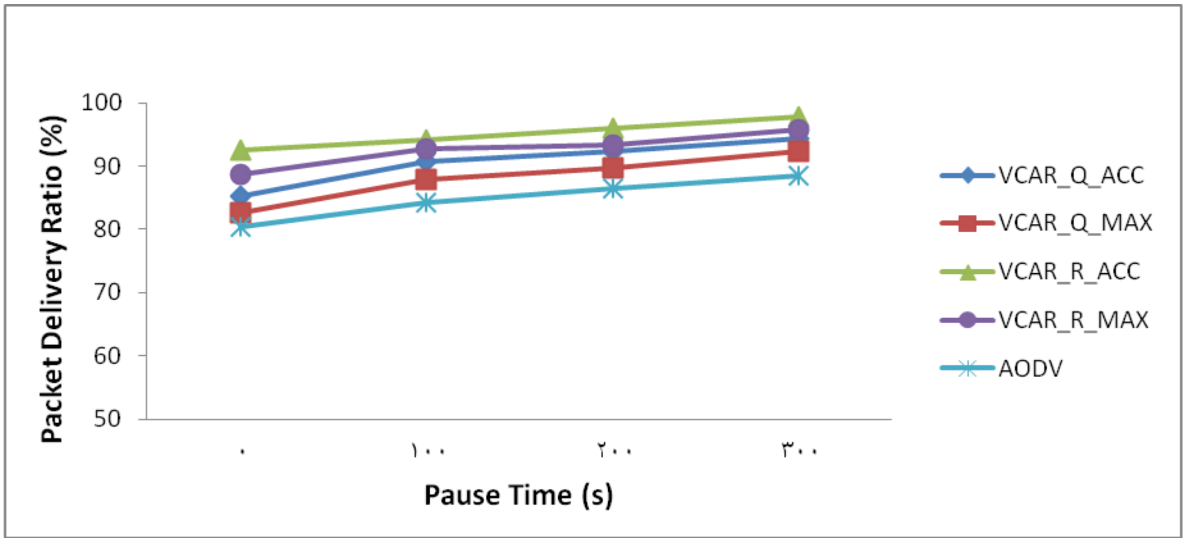


**Figure 5-2: Packet Delivery Ratio of 5 sources each one sends 2 packets/s**

Figure 5-3 shows the packet delivery ratio for all protocols when the number of sources is five and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 15.32, 10.60, 6.16, and 3.07 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 8.75, 6.45, 4.93, and 2.54 percent, respectively.

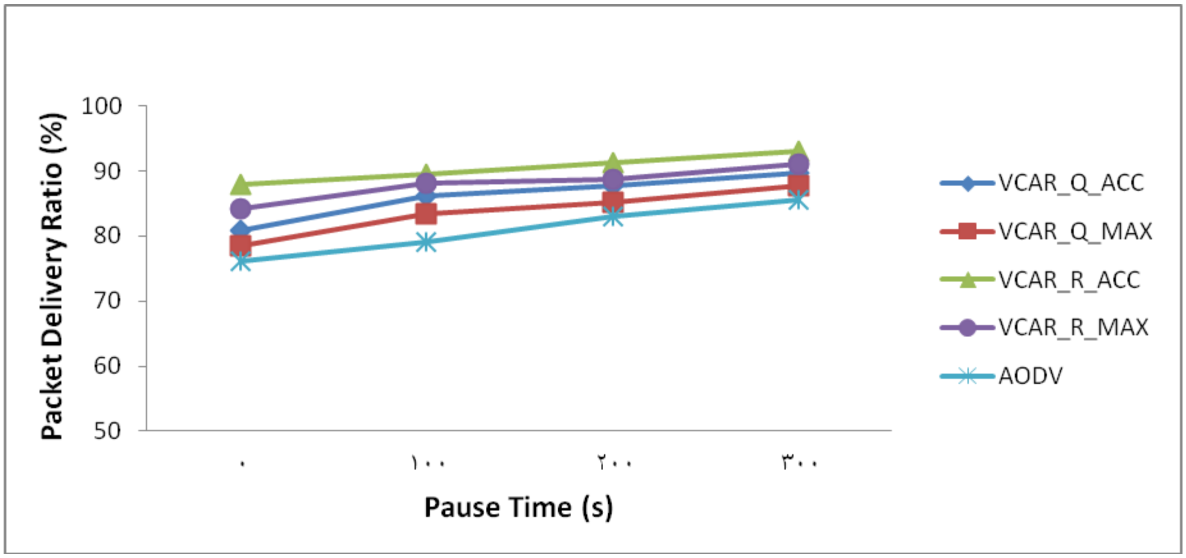


**Figure 5-3: Packet Delivery Ratio of 5 sources each one sends 4 packets/s**

Figure 5-4 shows the packet delivery ratio for all protocols when the number of sources is five and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 14.80, 10.19, 5.77, and 2.69 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 10.82, 8.47, 6.92, and 4.49 percent, respectively.

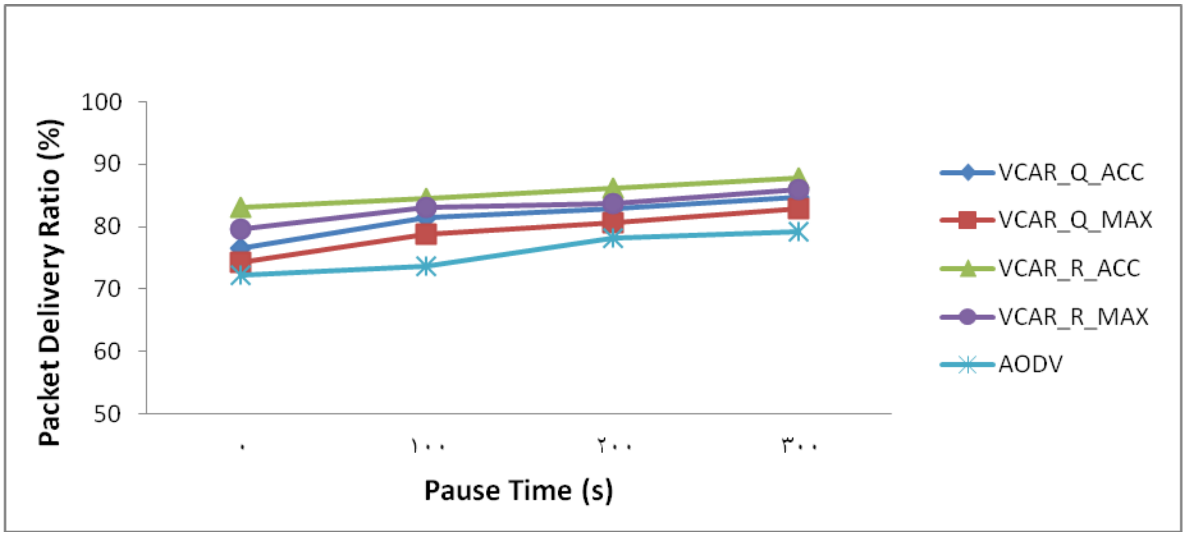


**Figure 5-4: Packet Delivery Ratio of 5 sources each one sends 6 packets/s**

Figure 5-5 shows the packet delivery ratio for all protocols when the number of sources is ten and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 14.73, 10.03, 5.62, and 2.54 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 6.14, 5.24, 3.74, and 1.37 percent, respectively.

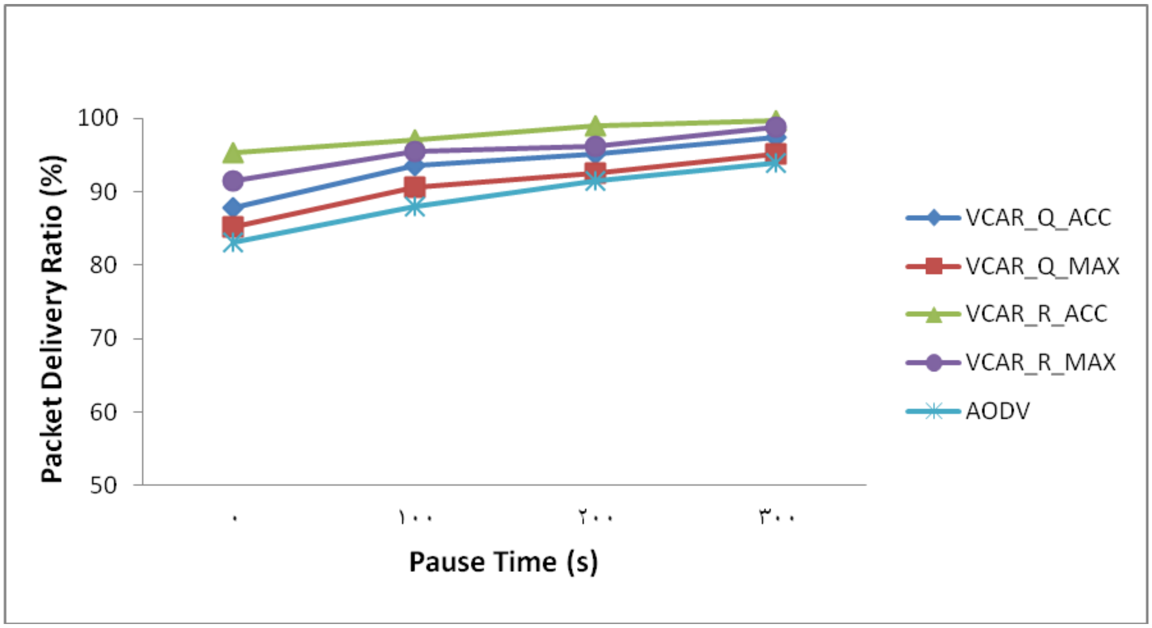


**Figure 5-5: Packet Delivery Ratio of 10 sources each one sends 1 packet/s**

Figure 5-6 shows the packet delivery ratio for all protocols when the number of sources is ten and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 22.43, 17.42, 12.71, and 4.29 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 10.24, 9.30, 7.74, and 5.28 percent, respectively.

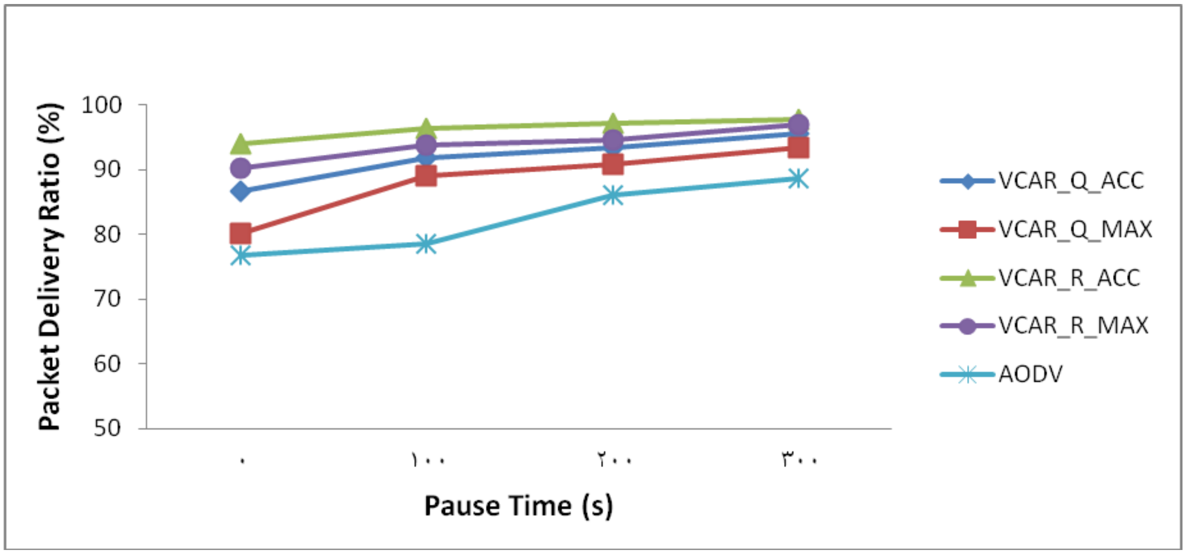


**Figure 5-6: Packet Delivery Ratio of 10 sources each one sends 2 packets/s**

Figure 5-7 shows the packet delivery ratio for all protocols when the number of sources is ten and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 28.26, 19.41, 14.66. and 3.19 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 23.19, 18.69, 15.79, and 11.12 percent, respectively.

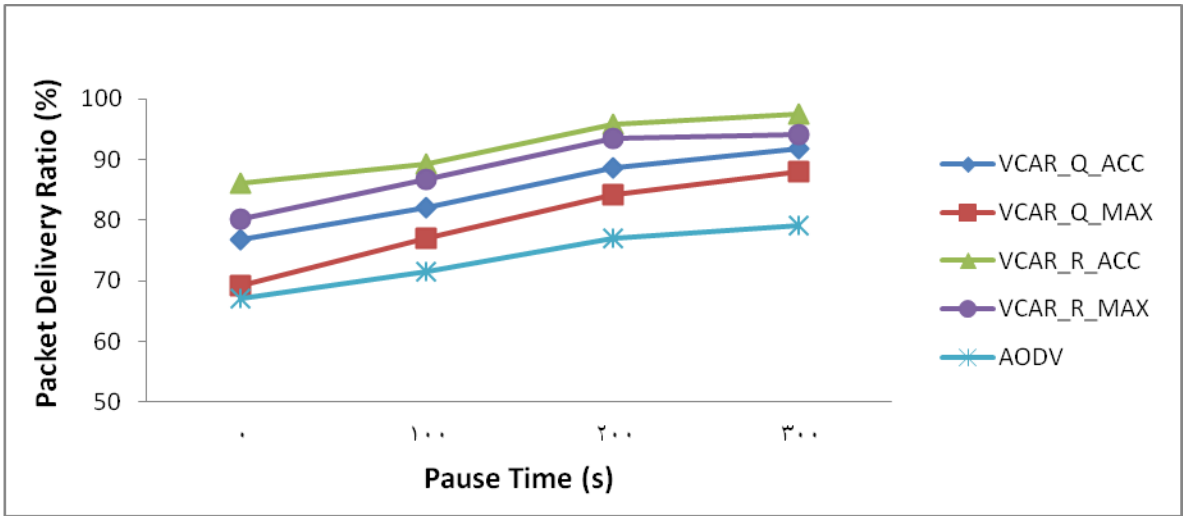


**Figure 5-7: Packet Delivery Ratio of 10 sources each one sends 4 packets/s**

Figure 5-8 shows the packet delivery ratio for all protocols when the number of sources is ten and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 31.35, 21.04, 20.03, and 5.46 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 26.69, 19.68, 16.39, and 11.11 percent, respectively.

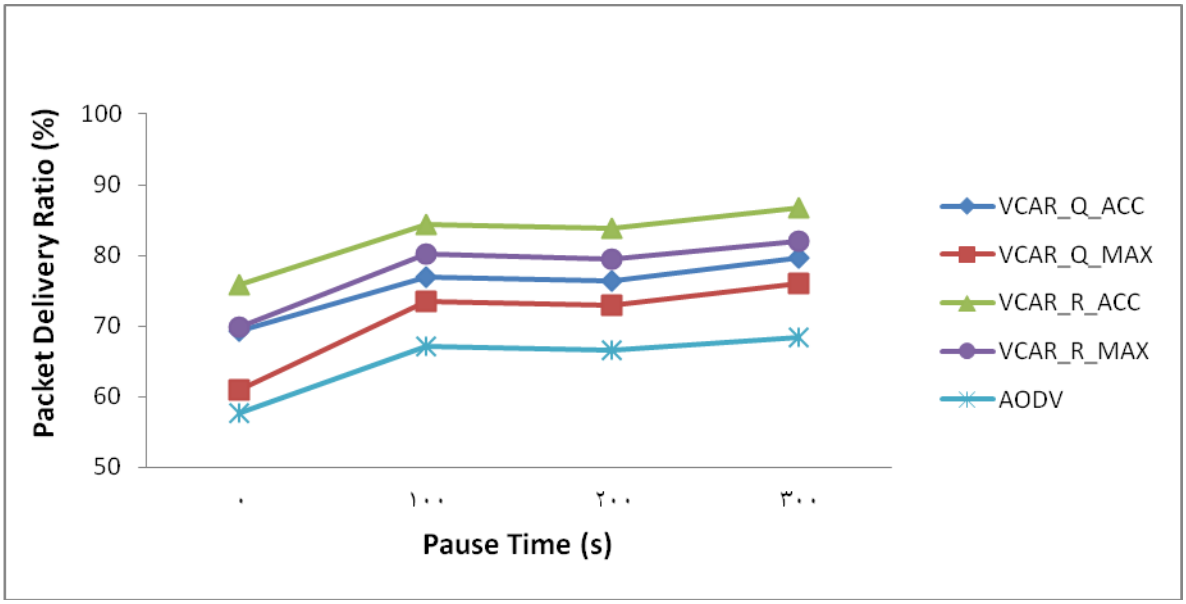


**Figure 5-8: Packet Delivery Ratio of 10 sources each one sends 6 packets/s**

Figure 5-9 shows the packet delivery ratio for all protocols when the number of sources is fifteen and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 22.36, 17.35, 12.64, and 4.23 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 18.84, 15.47, 14.72, and 10.67 percent, respectively.

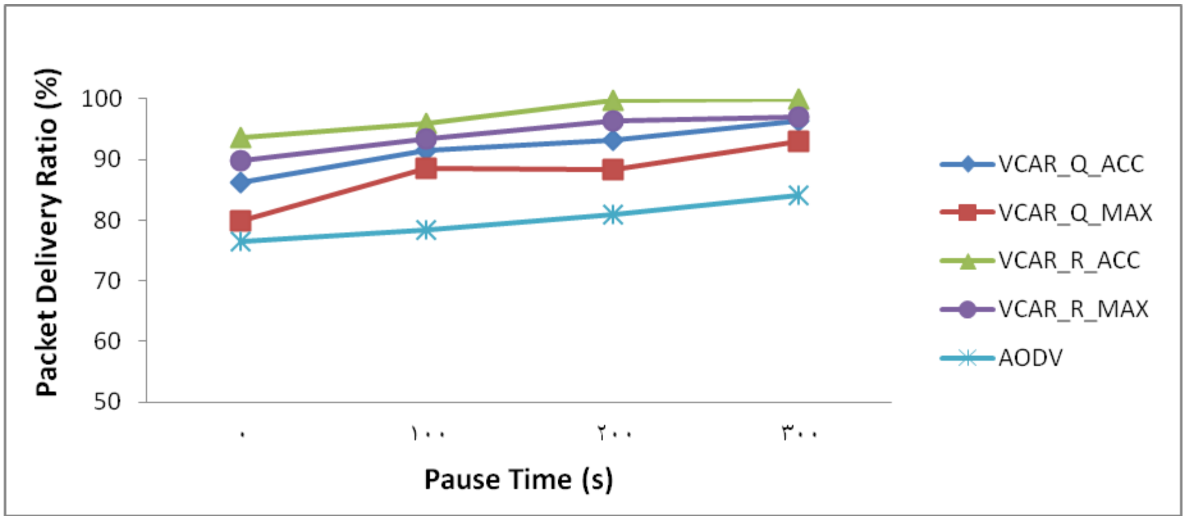


**Figure 5-9: Packet Delivery Ratio of 15 sources each one sends 1 packet/s**

Figure 5-10 shows the packet delivery ratio for all protocols when the number of sources is fifteen and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_MAX, and VCAR_Q_ACC outperform AODV by 33.71, 18.95, 18.29, and 2.80 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 26.57, 18.43, 15.54, and 10.87 percent, respectively.

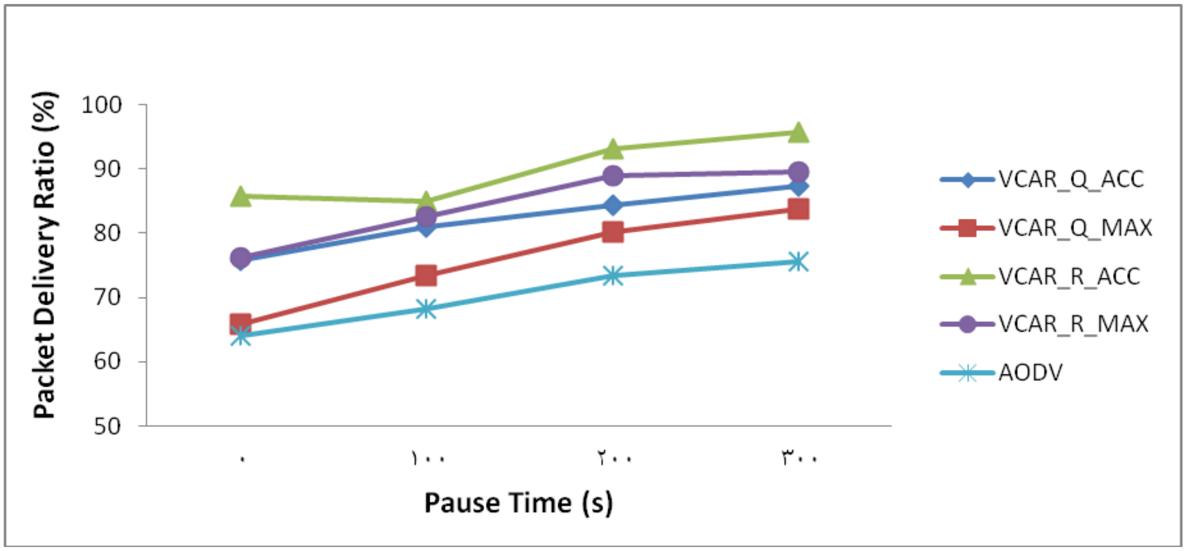


**Figure 5-10: Packet Delivery Ratio of 15 sources each one sends 2 packets/s**

Figure 5-11 shows the packet delivery ratio for all protocols when the number of sources is fifteen and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 30.90, 20.68, 19.73, and 5.25 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 26.72, 19.64, 16.35, and 11.07 percent, respectively.

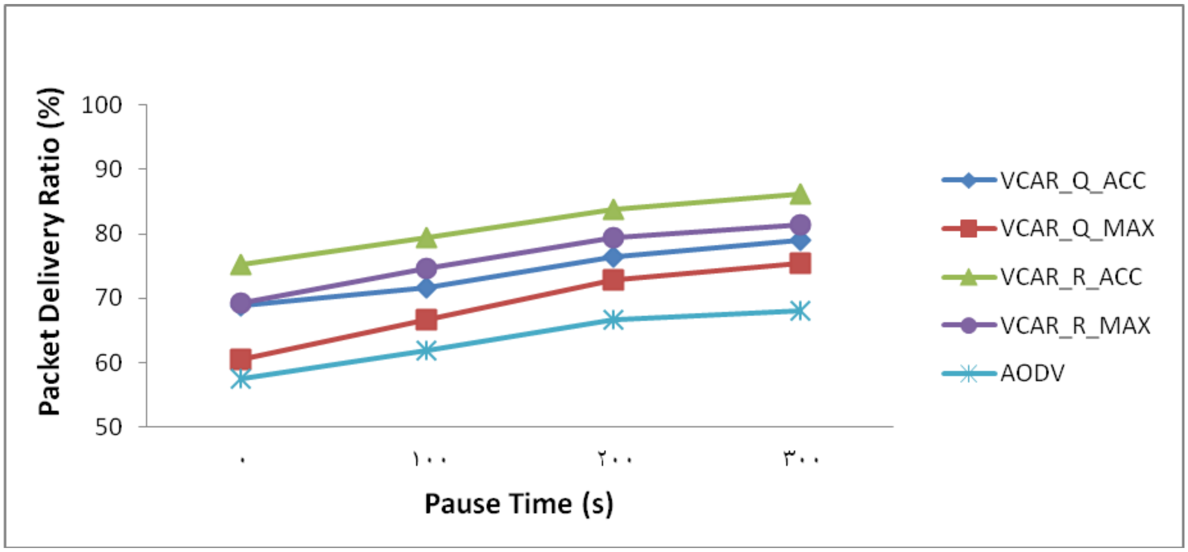


**Figure 5-11: Packet Delivery Ratio of 15 sources each one sends 4 packets/s**

Figure 5-12 shows the packet delivery ratio for all protocols when the number of sources is fifteen and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.00, 21.34, 18.24, and 6.80 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 25.73, 19.79, 16.12, and 10.27 percent, respectively.

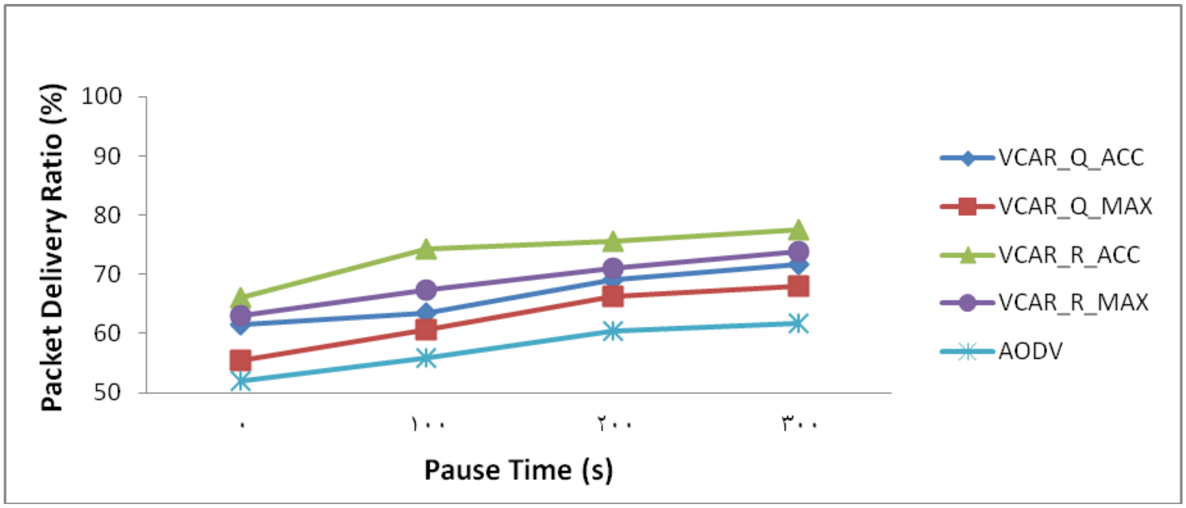


**Figure 5-12: Packet Delivery Ratio of 15 sources each one sends 6 packets/s**

## 5-2 Average End-to-End Delay.

Figures 5-13~5-24 show the average end-to-end delay for different transmission rate values and a different number of sources. The results also show that all VCAR protocols have a lower average end to end delay than the AODV protocol for all mobility levels, all number of sources, and all transmission rates. This is because AODV always uses the shortest route even if it is highly congested. While VCAR protocols use less congested routes. Another reason is that VCAR protocols use the most stable route, while AODV does not, so that the route selected by AODV is more likely to break than the route selected by VCAR protocols.

VCAR_R_ACC outperforms all other VCAR protocols in all the cases we have studied. On average, VCAR_R_ACC outperforms AODV by 28.86 percent for low mobility networks, and by 32.77 for high mobility networks.

Figure 5-13 shows the average end-to-end delay for all protocols when the number of sources is five and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 32.25, 26.97, 18.11, and 11.03 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 31.04, 26.85, 26.34, and 21.55 percent, respectively.

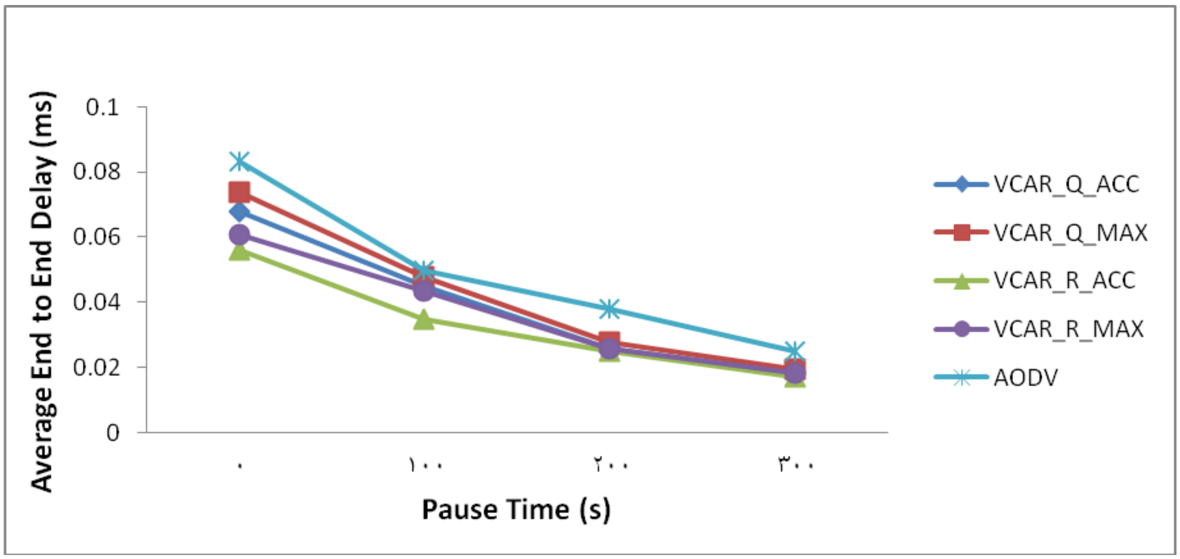


**Figure 5-13: Average End to End Delay of 5 sources each one sends 1 packet/s**

Figure 5-14 shows the average end-to-end delay for all protocols when the number of sources is five and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 28.87, 23.31, 14.01, and 6.59 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 29.93, 25.67, 25.15, and 20.29 percent, respectively.

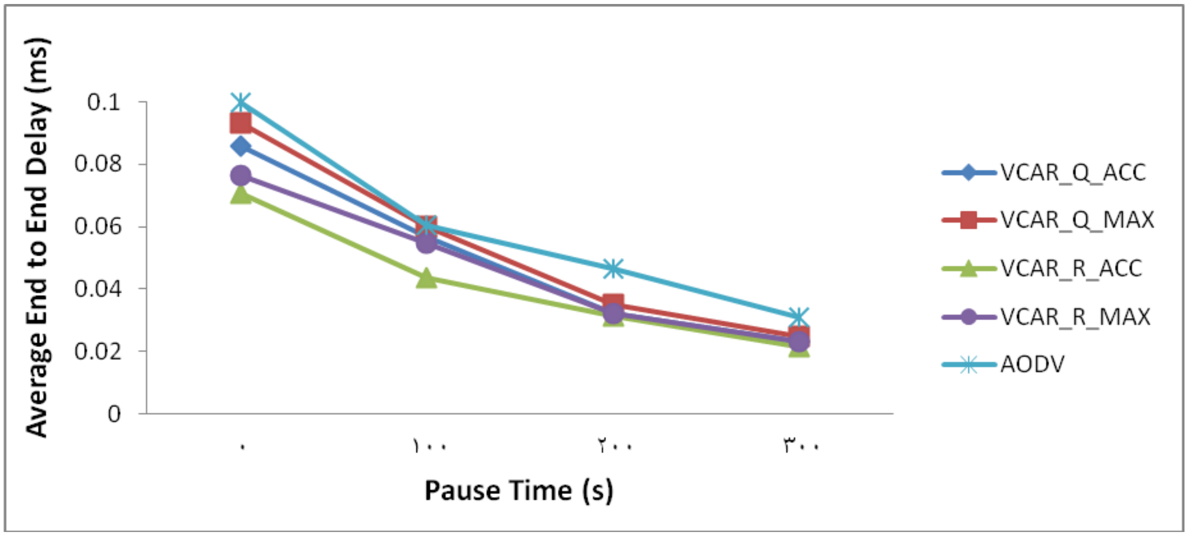


**Figure 5-14: Average End to End Delay of 5 sources each one sends 2 packets/s**

Figure 5-15 shows the average end-to-end delay for all protocols when the number of sources is five and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 28.36, 22.77, 13.40, and 5.93 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.78, 23.39, 22.86, and 17.84 percent, respectively.

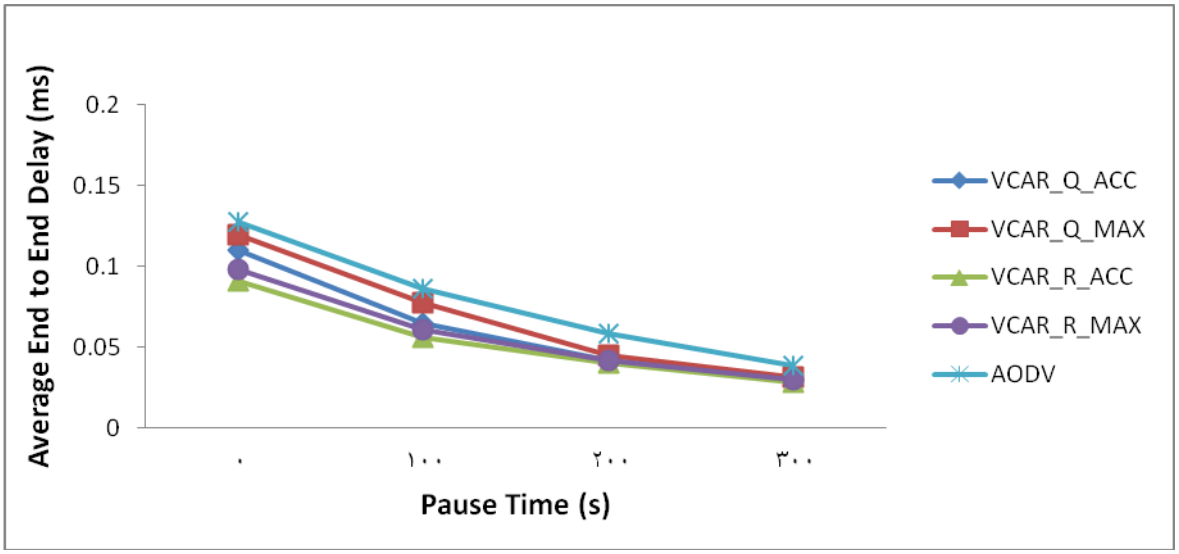


**Figure 5-15: Average End to End Delay of 5 sources each one sends 4 packets/s**

Figure 5-16 shows the average end-to-end delay for all protocols when the number of sources is five and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.80, 22.17, 12.73, and 5.20 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.14, 22.71, 22.18, and 17.11 percent, respectively.

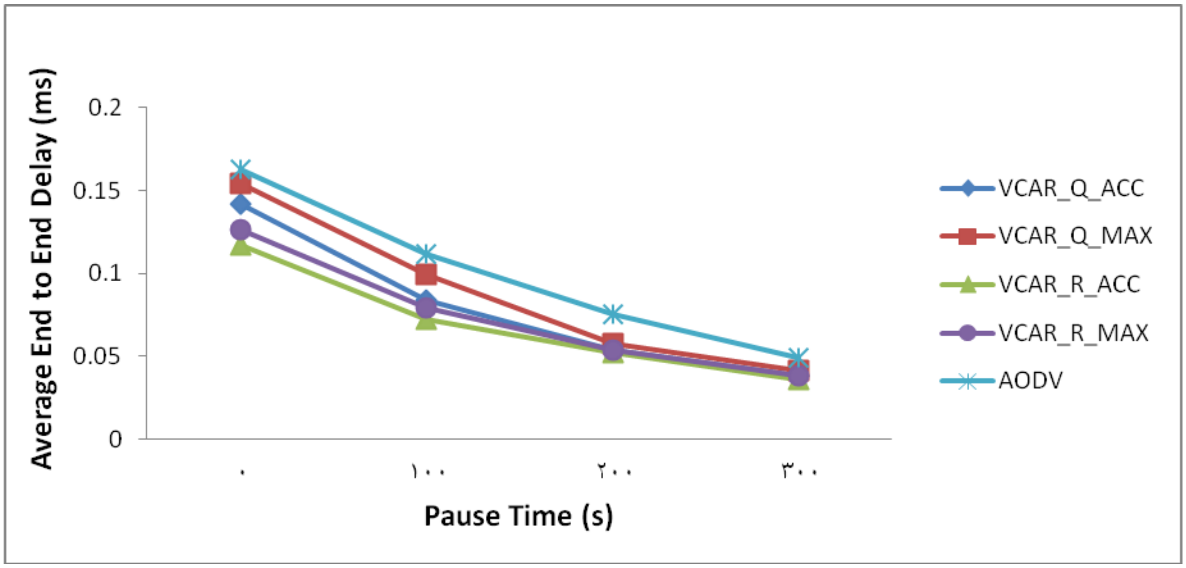


**Figure 5-16: Average End to End Delay of 5 sources each one sends 6 packets/s**

Figure 5-17 shows the average end-to-end delay for all protocols when the number of sources is ten and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 30.84, 25.44, 16.40, and 9.18 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 33.91, 29.90, 29.41, and 24.82 percent, respectively.

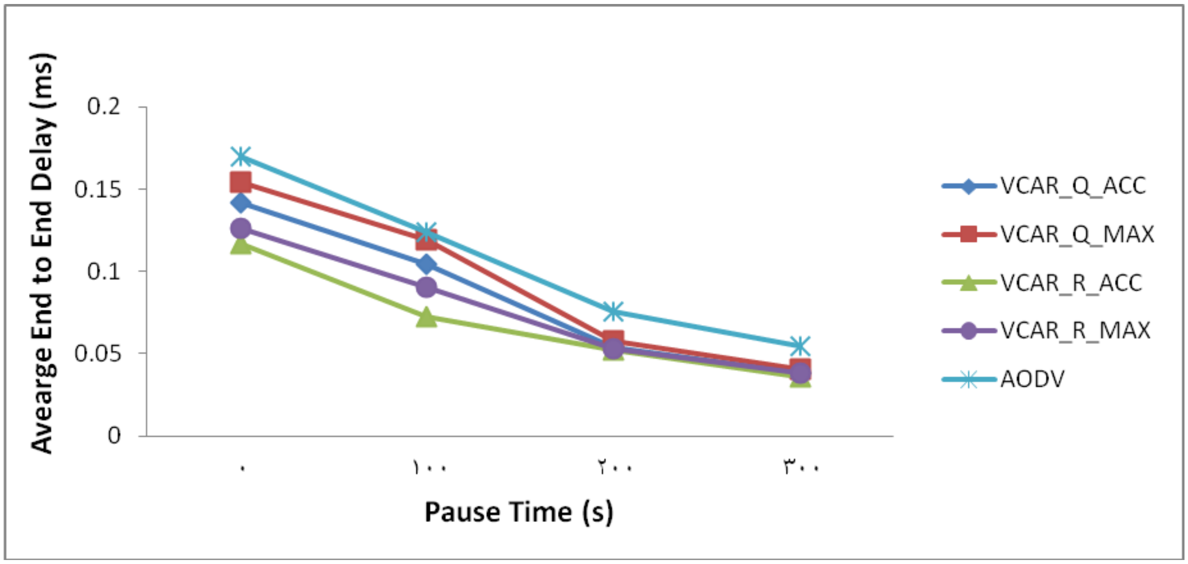


**Figure 5-17: Average End to End Delay of 10 sources each one sends 1 packet/s**

Figure 5-18 shows the average end-to-end delay for all protocols when the number of sources is ten and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 26.52, 20.78, 11.17, and 3.51 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 22.90, 18.21, 17.64, and 12.29 percent, respectively.

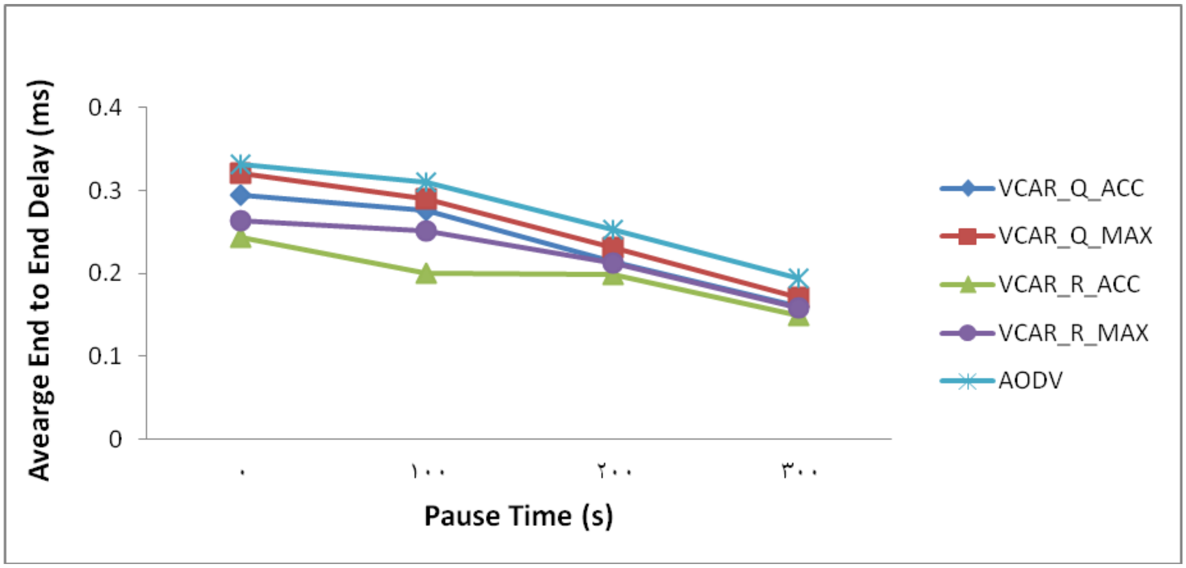


**Figure 5-18: Average End to End Delay of 10 sources each one sends 2 packets/s**

Figure 5-19 shows the average end-to-end delay for all protocols when the number of sources is ten and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.10, 23.45, 15.27, and 6.76 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 28.51, 22.60, 20.33, and 14.22 percent, respectively.

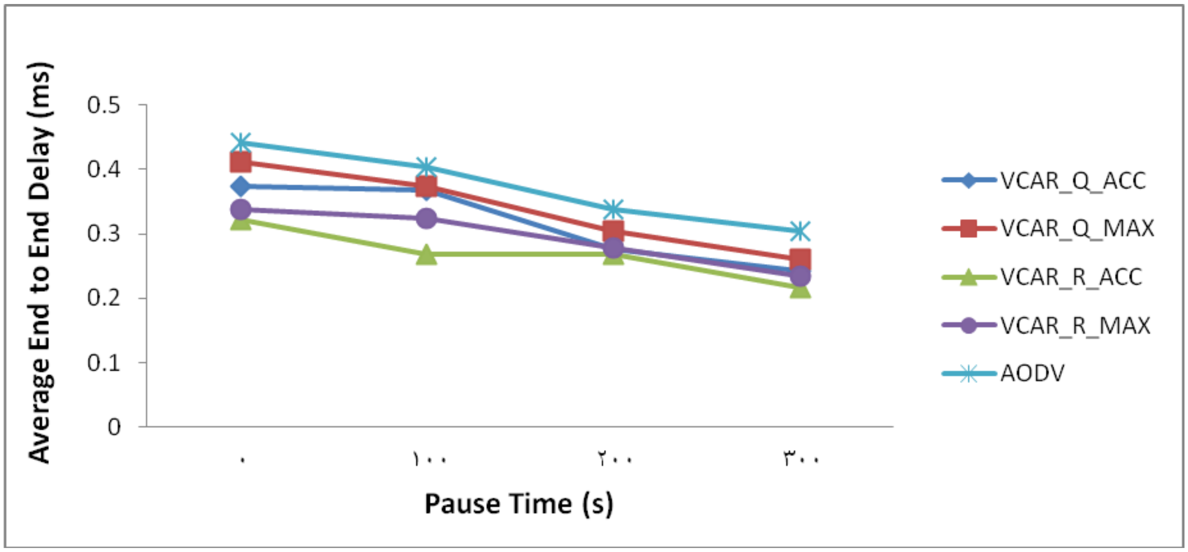


**Figure 5-19: Average End to End Delay of 10 sources each one sends 4 packets/s**

Figure 5-20 shows the average end-to-end delay for all protocols when the number of sources is ten and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 35.03, 32.79, 25.73, and 16.30 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 33.29, 25.88, 21.55, and 14.12 percent, respectively.

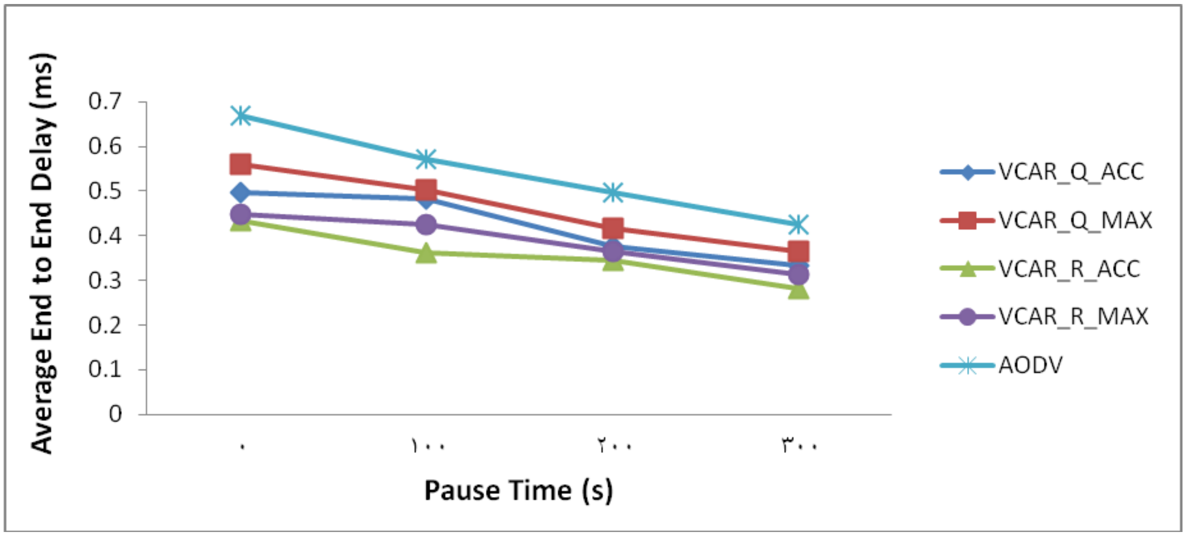


**Figure 5-20: Average End to End Delay of 10 sources each one sends 6 packets/s**

Figure 5-21 shows the average end-to-end delay for all protocols when the number of sources is fifteen and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 34.37, 29.25, 20.66, and 13.81 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 23.43, 18.77, 18.21, and 12.89 percent, respectively.

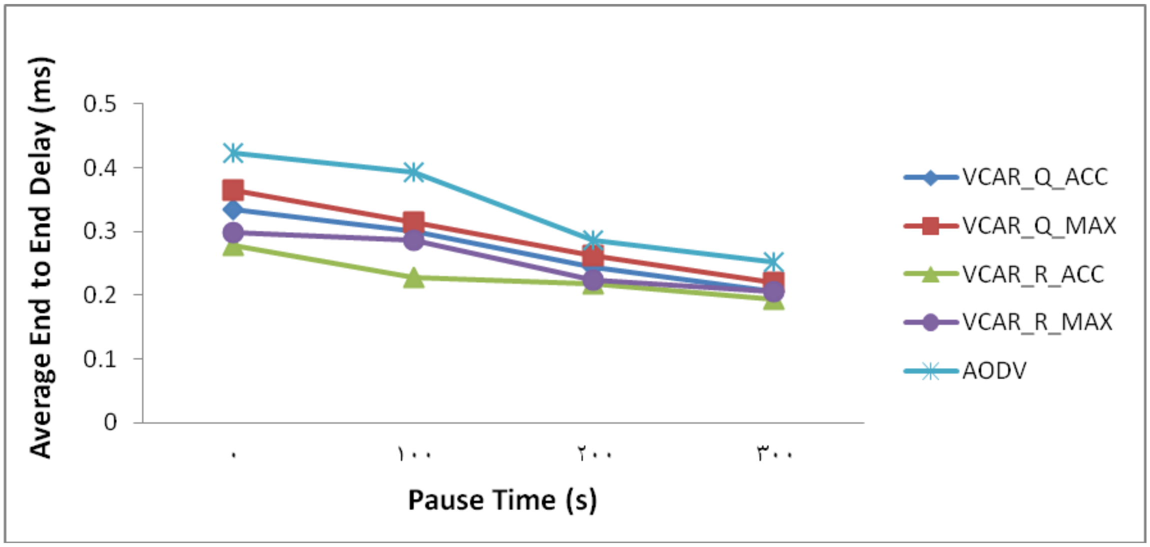


**Figure 5-21: Average End to End Delay of 15 sources each one sends 1 packet/s**

Figure 5-22 shows the average end-to-end delay for all protocols when the number of sources is fifteen and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 32.97, 29.61, 22.09, and 14.26 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.69, 21.41, 19.41, and 13.23 percent, respectively.

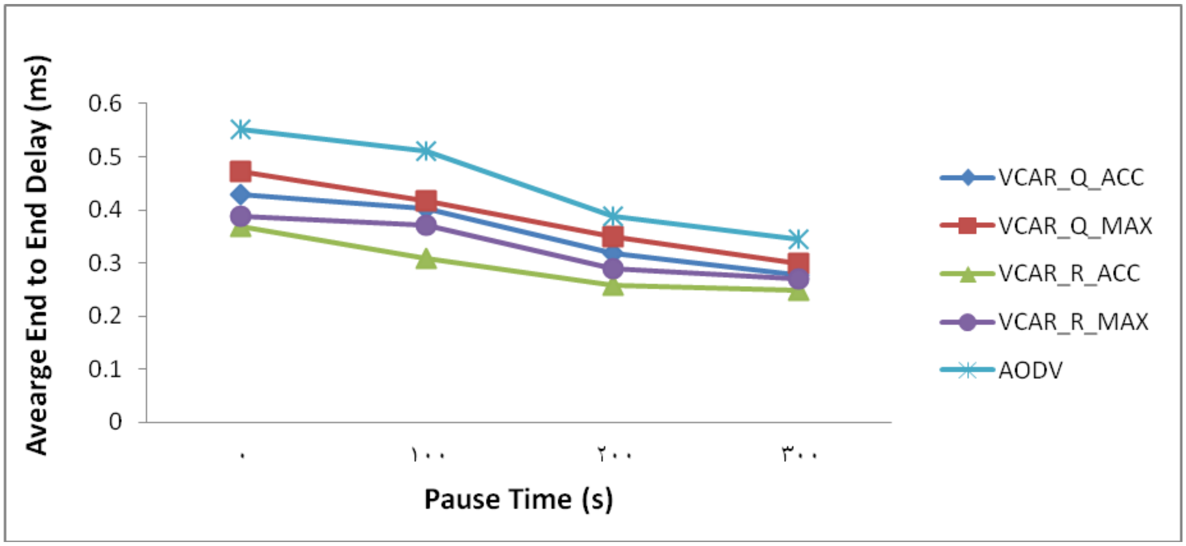


**Figure 5-22: Average End to End Delay of 15 sources each one sends 2 packets/s**

Figure 5-23 shows the average end-to-end delay for all protocols when the number of sources is fifteen and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 34.14, 30.53, 24.08, and 15.38 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 31.77, 24.62, 20.67, 13.64 percent, respectively.

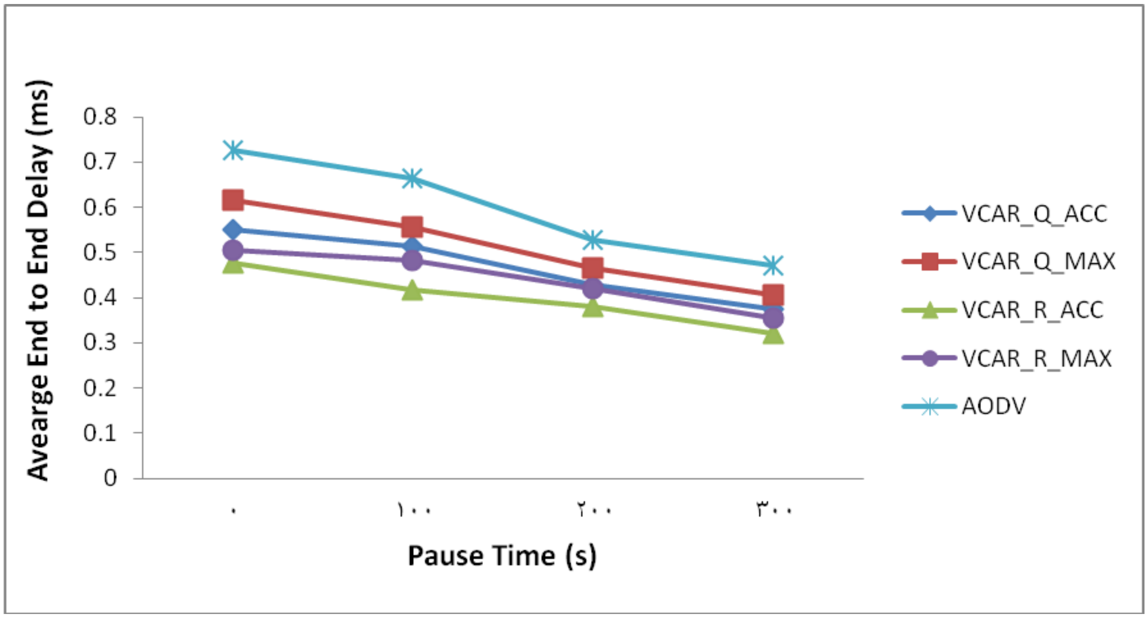


**Figure 5-23: Average End to End Delay of 15 sources each one sends 4 packets/s**

Figure 5-24 shows the average end-to-end delay for all protocols when the number of sources is fifteen and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 36.32, 32.33, 27.00, and 17.57 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 28.24, 23.68, 19.80, and 13.91 percent, respectively.

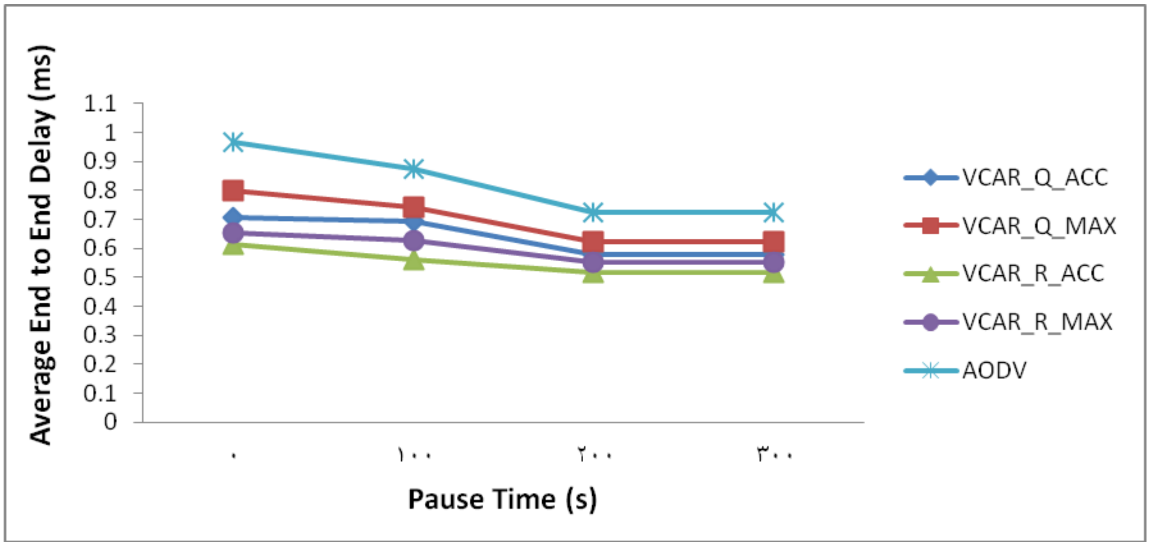


**Figure 5-24: Average End to End Delay of 15 sources each one sends 6 packets/s**

## 5-3 Routing Overhead

Figures 5-25~5-36 show the routing overhead for all the simulation scenarios. The results show that all VCAR protocols have a higher routing overhead than AODV especially in high mobility scenarios. In VCAR, the destination may send multiple replies, because it always searches for the best path among all available paths. This behavior increases the number of RREP packets, and consequently increases the routing overhead.

Figure 5-25 shows the routing overhead for all protocols when the number of sources is five and each source sends one packet per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 2.70, 3.64, 6.00, 6.43 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms

VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 4.28, 5.17, 8.60, 11.20 percent, respectively.

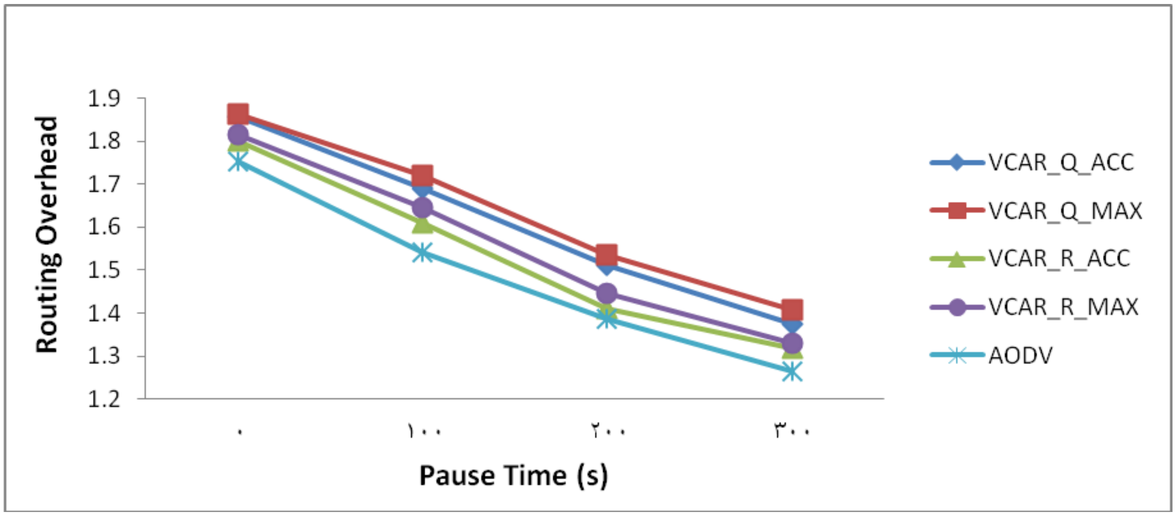


**Figure 5-25: Routing Overhead of 5 sources each one sends 1 packet/s**

Figure 5-26 shows the routing overhead for all protocols when the number of sources is five and each source sends two packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 2.89, 3.83, 6.20, and 6.62 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 4.38, 5.28, 8.71, and 11.31 percent, respectively.

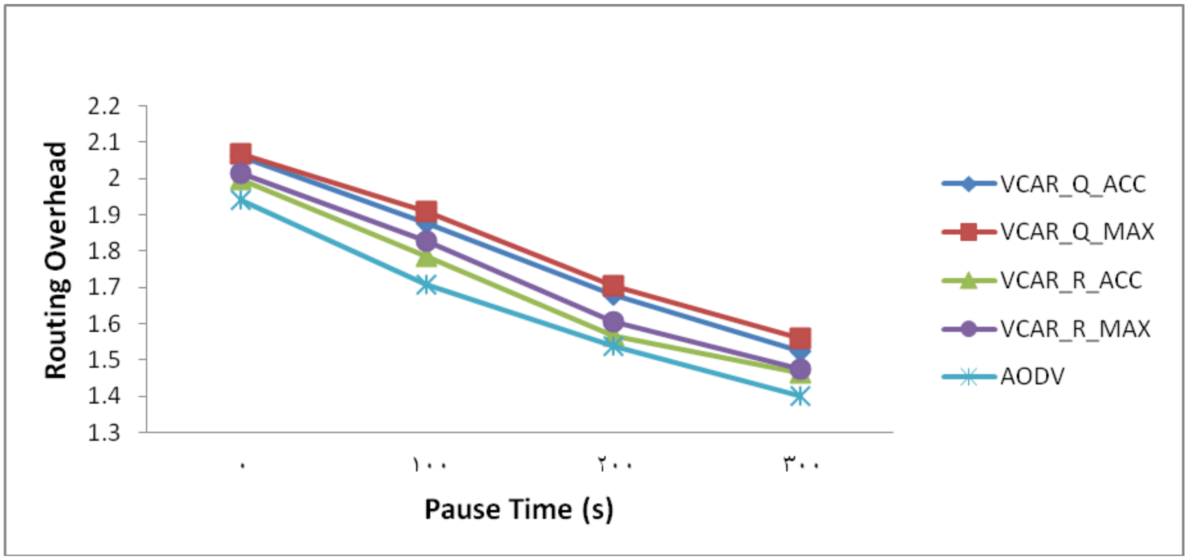


**Figure 5-26: Routing Overhead of 5 sources each one sends 2 packets/s**

Figure 5-27 shows the routing overhead for all protocols when the number of sources is five and each source sends four packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.00, 3.95, 6.31, and 6.74 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.30, 4.18, 7.58, and 10.16 percent, respectively.

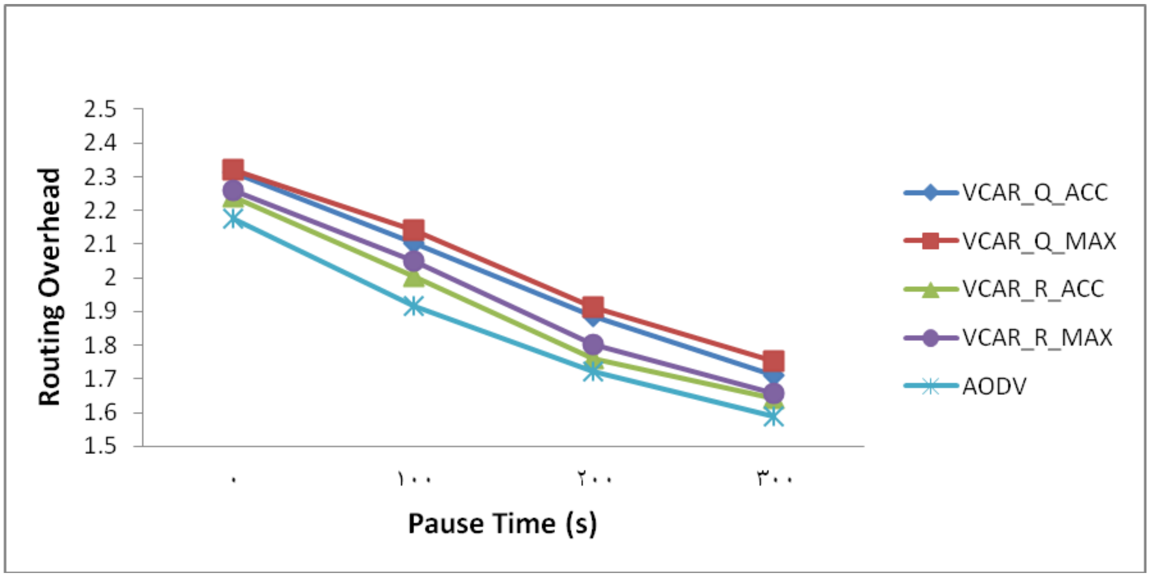


**Figure 5-27: Routing Overhead of 5 sources each one sends 4 packets/s**

Figure 5-28 shows the routing overhead for all protocols when the number of sources is five and each source sends six packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.12, 4.07, 6.44, and 6.86 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.04, 3.92, 7.32, and 9.89 percent, respectively.

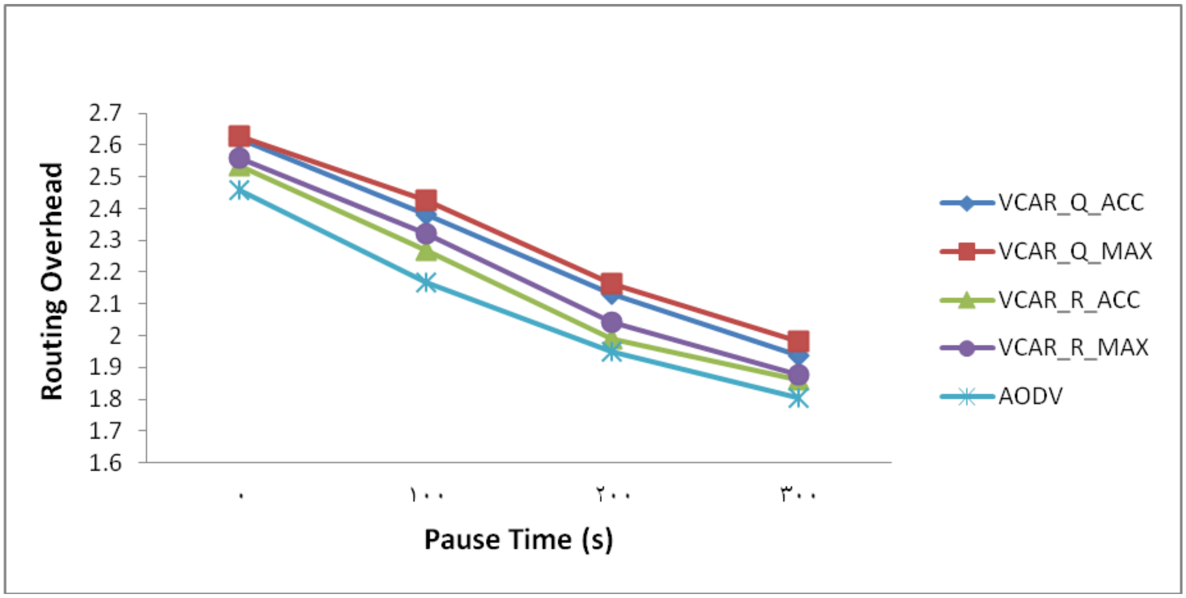


**Figure 5-28: Routing Overhead of 5 sources each one sends 6 packets/s**

Figure 5-29 shows the routing overhead for all protocols when the number of sources is ten and each source sends one packet per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 11.00, 12.02, 14.28, and 15.03 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.56, 4.47, 7.88, and 10.46 percent, respectively.

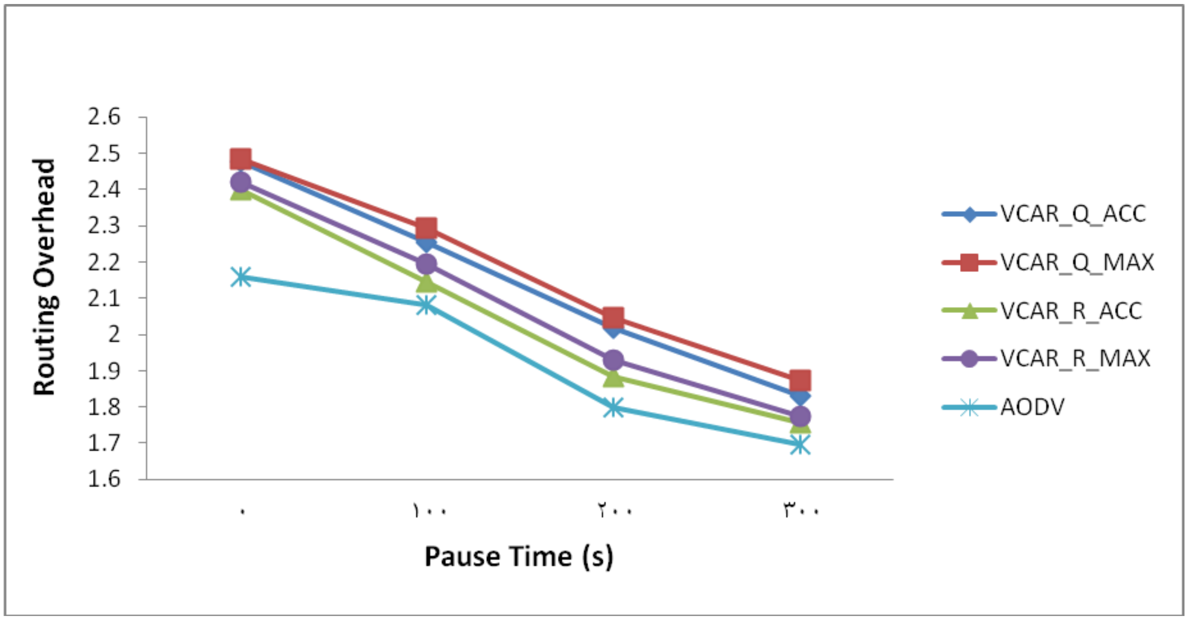


**Figure 5-29: Routing Overhead of 10 sources each one sends 1 packet/s**

Figure 5-30 shows the routing overhead for all protocols when the number of sources is ten and each source sends two packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 13.56, 14.59, 17.21, and 17.67 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.06, 1.92, 5.27, and 7.81 percent, respectively.

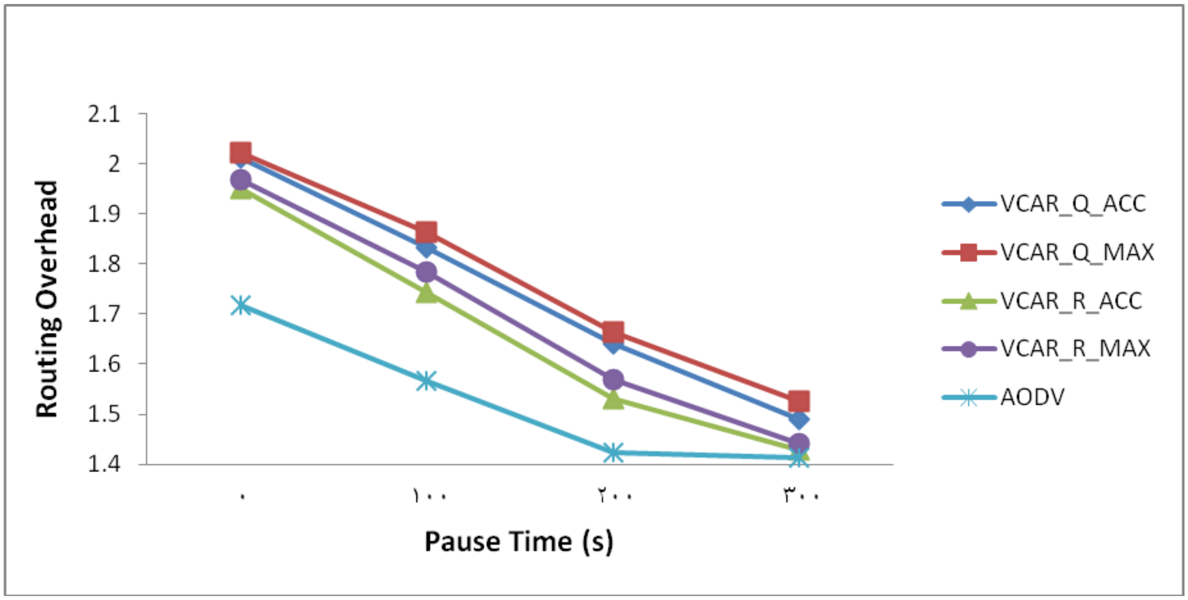


**Figure 5-30: Routing Overhead of 10 sources each one sends 2 packets/s**

Figure 5-31 shows the routing overhead for all protocols when the number of sources is ten and each source sends four packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.17, 2.11, 4.47, and 4.90 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.97, 3.53, 6.84, and 9.36 percent, respectively.

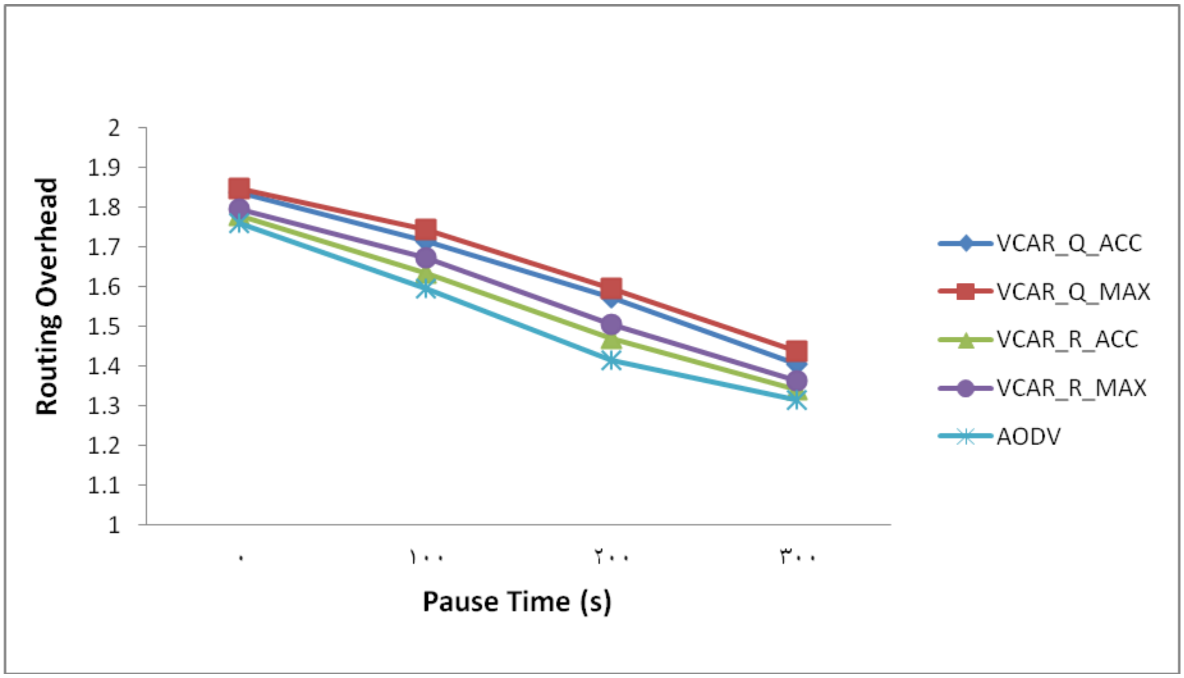


**Figure 5-31: Routing Overhead of 10 sources each one sends 4 packets/s**

Figure 5-32 shows the routing overhead for all protocols when the number of sources is ten and each source sends six packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.87, 6.52, 12.84, and 14.00 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.96, 10.79, 19.92, and 26.92 percent, respectively.

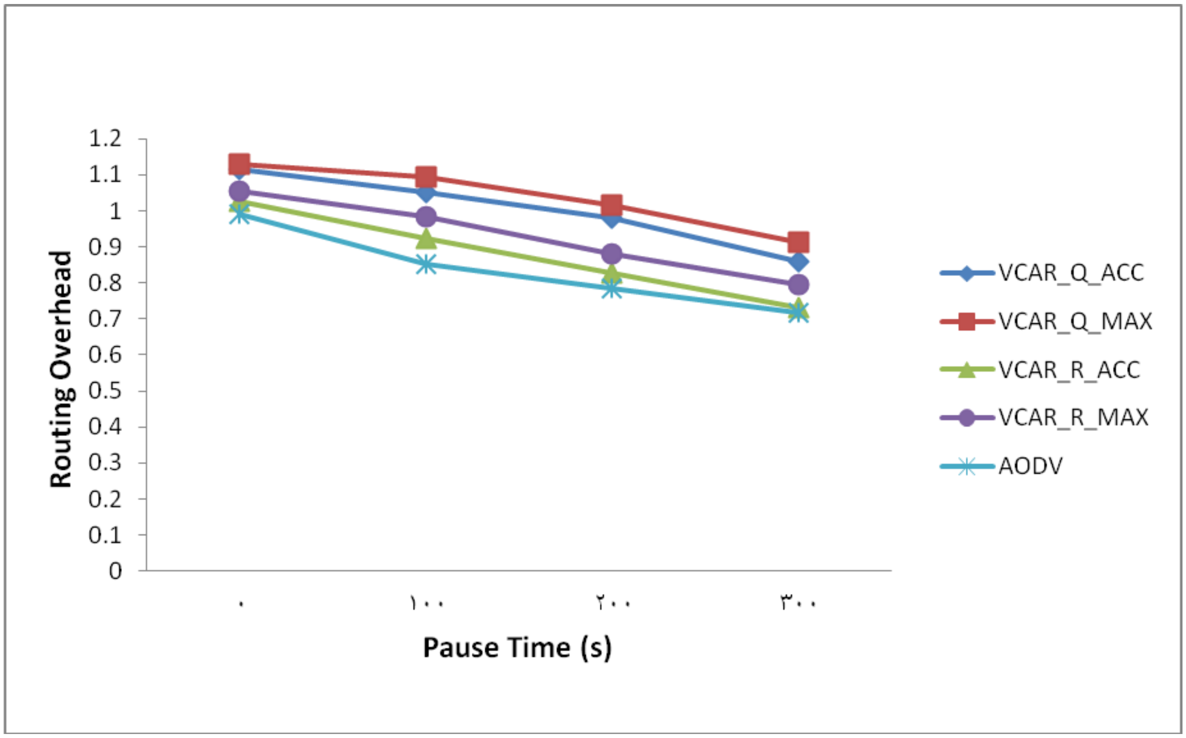


**Figure 5-32: Routing Overhead of 10 sources each one sends 6 packets/s**

Figure 5-33 shows the routing overhead for all protocols when the number of sources is fifteen and each source sends one packet per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.60, 2.53, 4.87, and 5.29 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 0.35, 1.92, 5.25, and 7.77 percent, respectively.

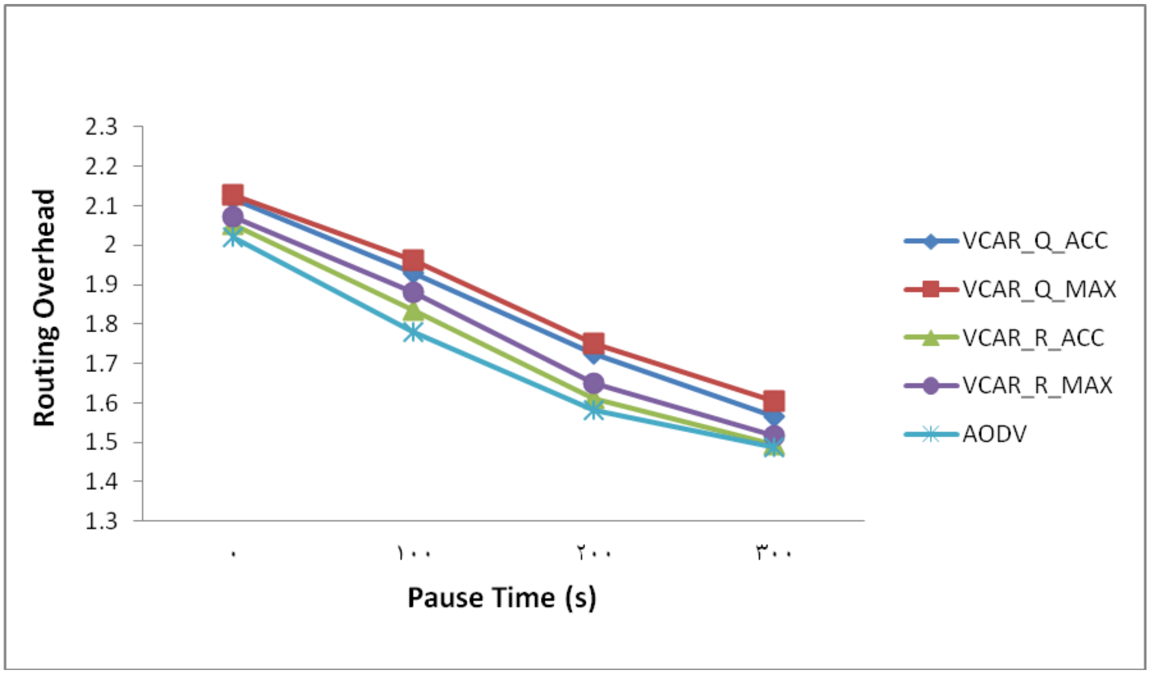


**Figure 5-33: Routing Overhead of 15 sources each one sends 1 packet/s**

Figure 5-34 shows the routing overhead for all protocols when the number of sources is fifteen and each source sends two packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.63, 2.57, 4.94, and 5.37 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 0.34, 1.88, 5.14, and 7.62 percent, respectively.

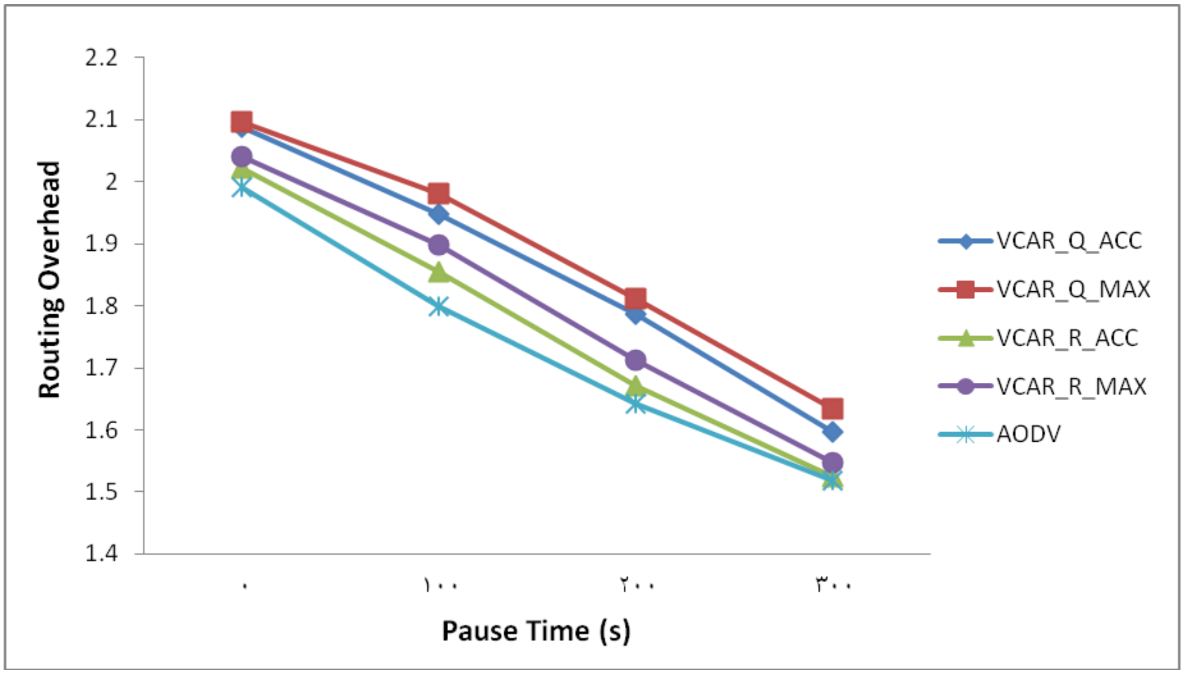


**Figure 5-34: Routing Overhead of 15 sources each one sends 2 packets/s**

Figure 5-35 shows the routing overhead for all protocols when the number of sources is fifteen and each source sends four packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 3.4, 6.05, 12.38, and 13.66 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 1.81, 10.63, 19.80, and 26.92 percent, respectively.

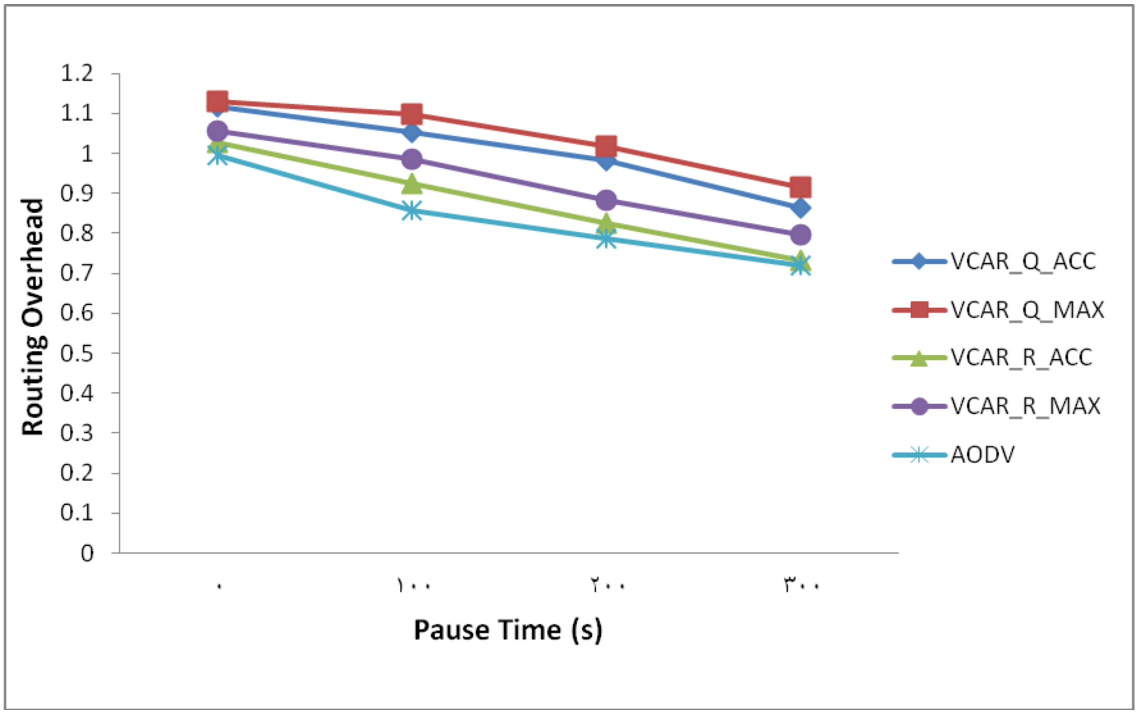


**Figure 5-35: Routing Overhead of 15 sources each one sends 4 packets/s**

Figure 5-36 shows the routing overhead for all protocols when the number of sources is fifteen and each source sends six packets per second. The figure shows that AODV outperforms all VCAR protocols for all pause times. When the pause time is equal to zero (high mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 2.16, 4.92, 11.19, and 12.60 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), AODV outperforms VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX by 4.39, 13.58, 22.99, and 30.48 percent, respectively.

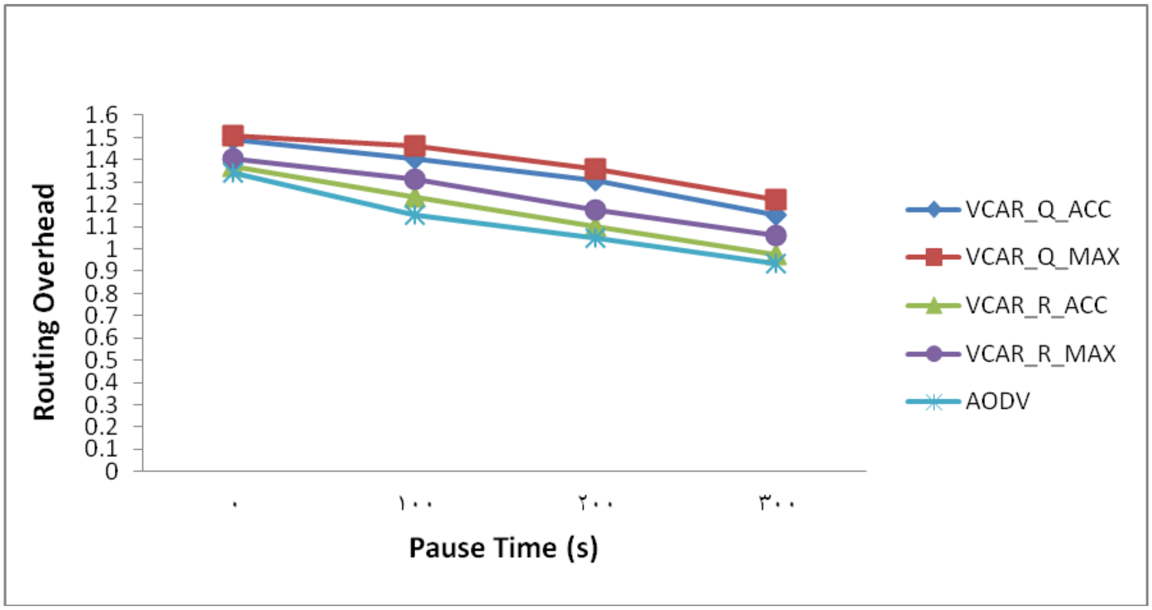


**Figure 5-36: Routing Overhead of 15 sources each one sends 6 packets/s**

## 5-4 Energy Consumption Percentage

Figures 5-37~5-48 show the energy percentage consumption for all simulation scenarios. We can see from these figures that all VCAR protocols outperform AODV in terms of power consumption percentage for all mobility levels, all number of sources , and all transmission rates. This is because VCAR maintains the nodes' energy as much as possible by distributing the load among the intermediate nodes. It reduces the probability of the intermediate node to be exhausted, and subsequently reduces the link failures. This mechanism used by VCAR reduces the power consumption needed to do route maintenance.

Figure 5-37 shows the energy consumption percentage for all protocols when the number of sources is five and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 29.37, 27.24, 21.75, and 19.06 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 30.81, 28.33, 24.07, and 18.48 percent, respectively.

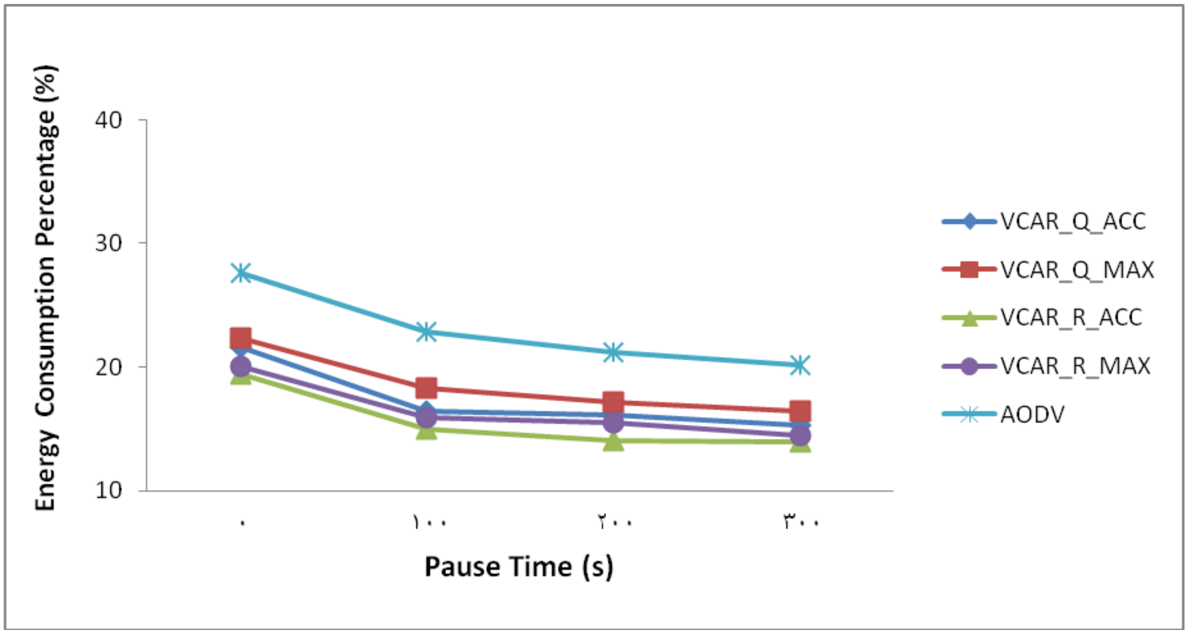


**Figure 5-37: Energy Consumption Percentage of 5 sources each one sends 1 packet/s**

Figure 5-38 shows the energy consumption percentage for all protocols when the number of sources is five and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.38, 25.40, 23.66, and 17.77 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 28.03, 25.78, 21.90, and 16.82 percent, respectively.

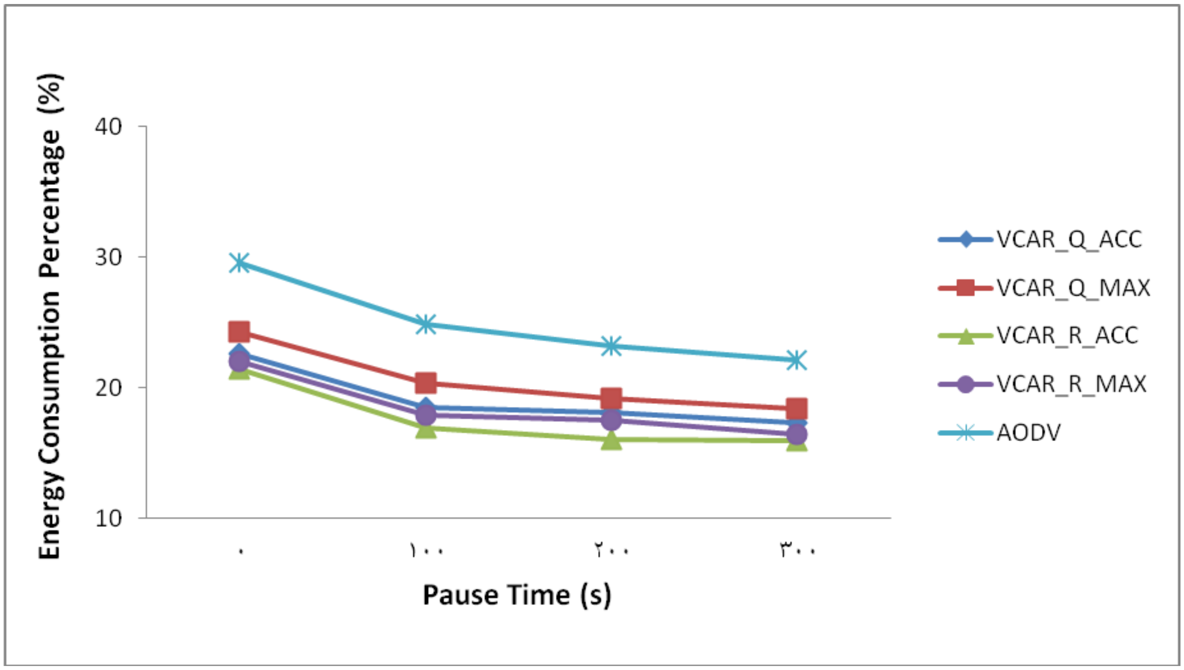


**Figure 5-38: Energy Consumption Percentage of 5 sources each one sends 2 packets/s**

Figure 5-39 shows the energy consumption percentage for all protocols when the number of sources is five and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 24.55, 22.28, 16.41, and 13.54 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.97, 25.40, 20.96, and 15.14 percent, respectively.

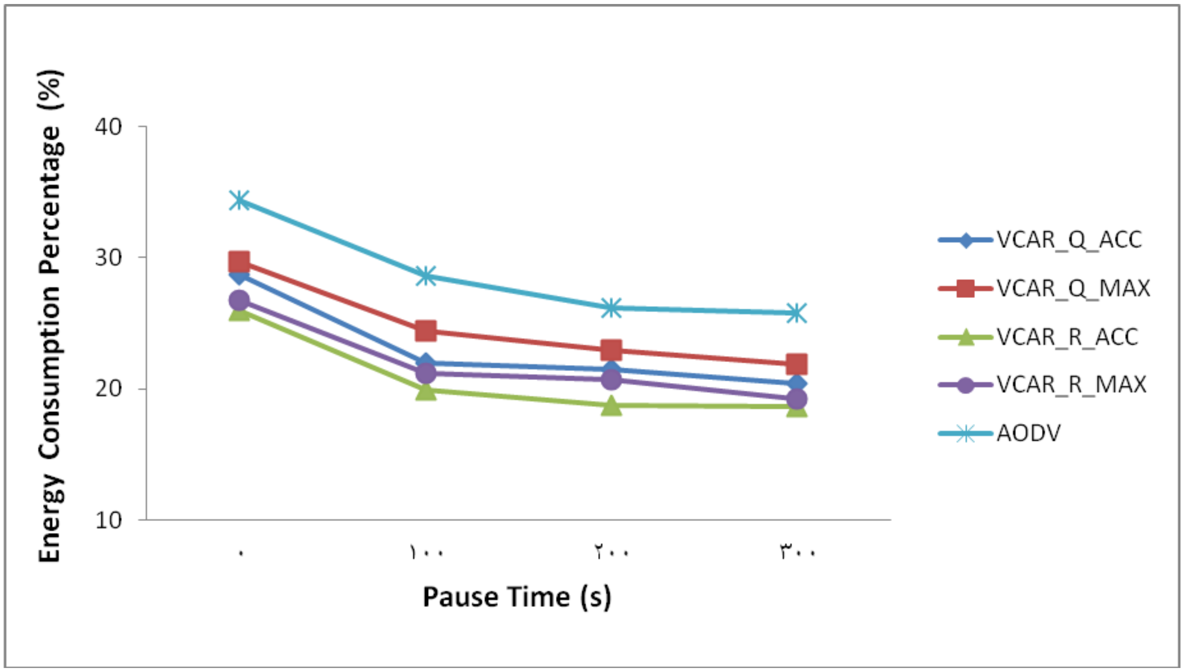


**Figure 5-39: Energy Consumption Percentage of 5 sources each one sends 4 packets/s**

Figure 5-40 shows the energy consumption percentage for all protocols when the number of sources is five and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 25.80, 23.56, 15.72, and 14.97 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 30.15, 27.64, 21.41, and 17.70 percent, respectively.

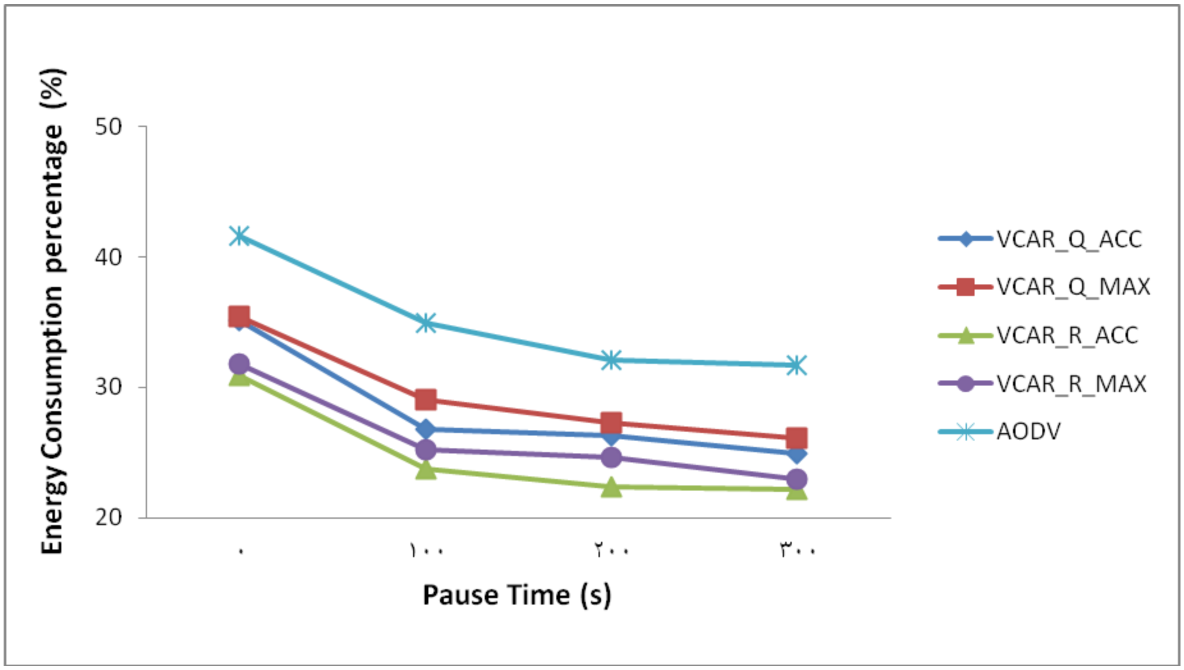


**Figure 5-40: Energy Consumption Percentage of 5 sources each one sends 6 packets/s**

Figure 5-41 shows the energy consumption percentage for all protocols when the number of sources is ten and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 27.56, 25.38, 19.75, and 16.99 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 35.25, 32.93, 28.94, and 23.71 percent, respectively.

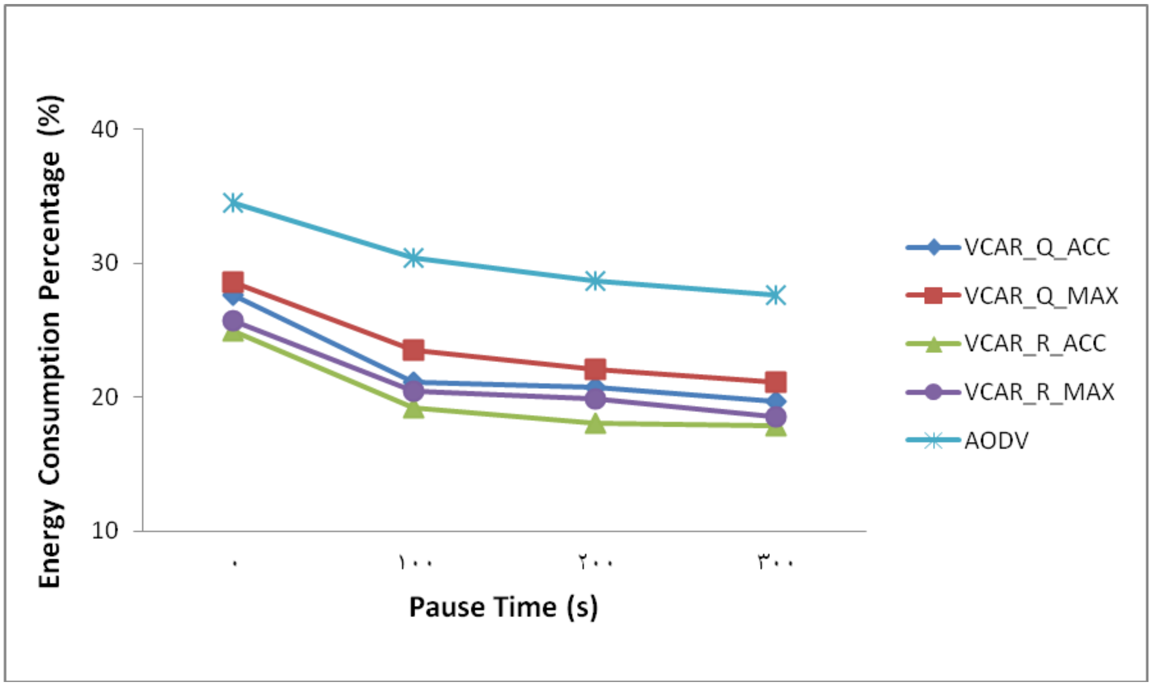


**Figure 5-41: Energy Consumption Percentage of 10 sources each one sends 1 packet/s**

Figure 5-42 shows the energy consumption percentage for all protocols when the number of sources is ten and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 24.91, 24.89, 21.50, 21.02 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 32.92, 32.53, 30.53, and 27.46 percent, respectively.

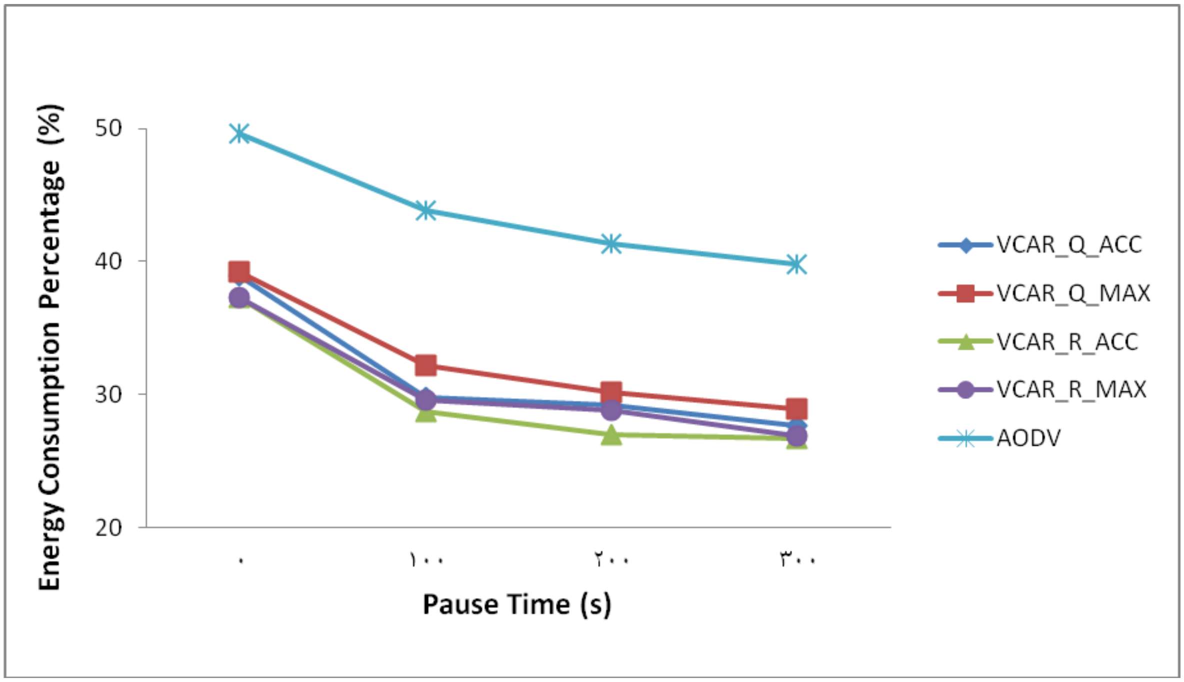


**Figure 5-42:Energy Consumption Percentage of 10 sources each one sends 2 packets/s**

Figure 5-43 shows the energy consumption percentage for all protocols when the number of sources is ten and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 16.80, 14.30, 12.20, and 10.89 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by12.18, 7.89, 5.86, and 5.12 percent, respectively.

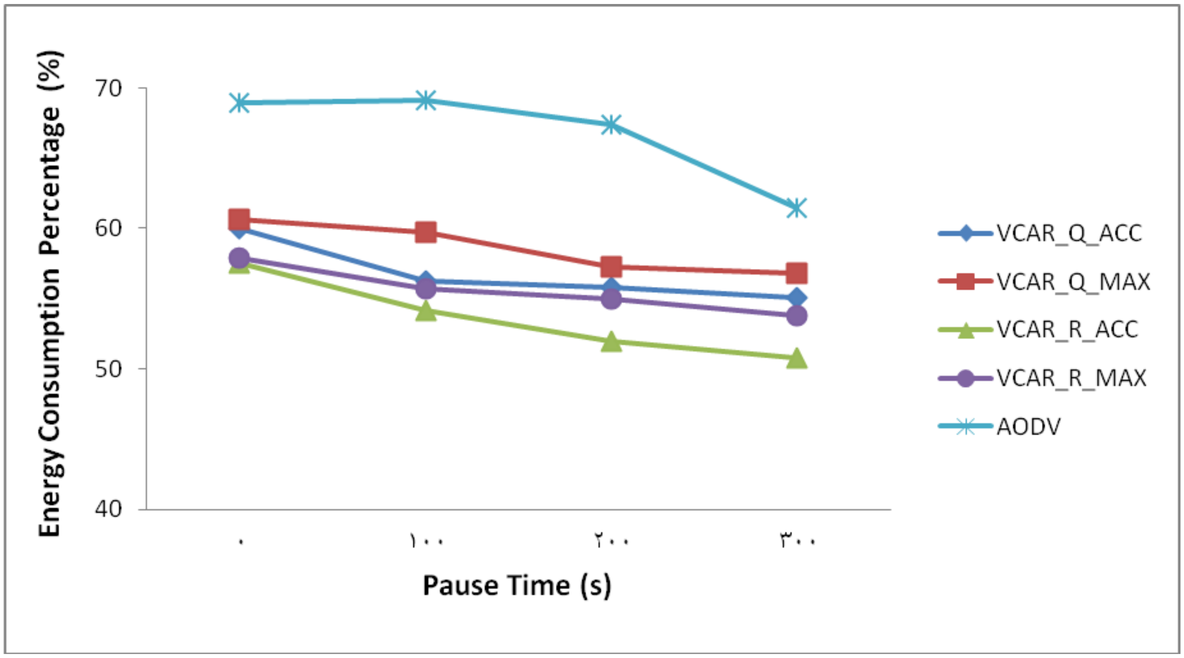


**Figure 5-43:Energy Consumption Percentage of 10 sources each one sends 4 packets/s**

Figure 5-44 shows the energy consumption percentage for all protocols when the number of sources is ten and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 16.83, 15.41, 12.38, and 10.49 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 20.52, 20.06, 18.63, and 17.13 percent, respectively.

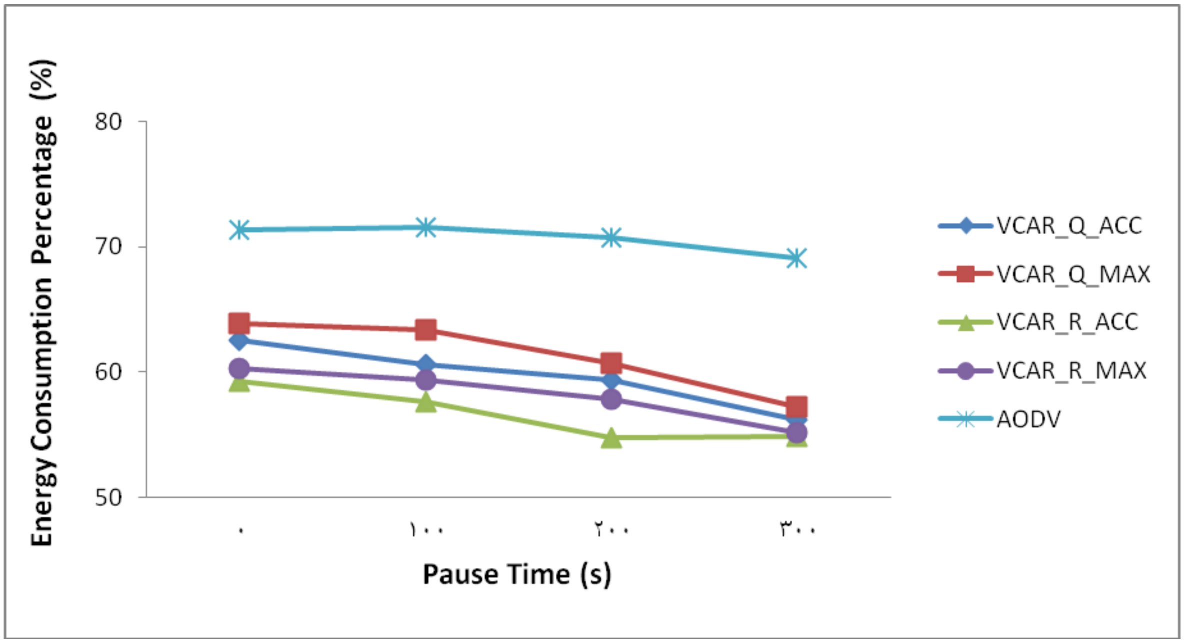


**Figure 5-44:Energy Consumption Percentage of 10 sources each one sends 6 packets/s**

Figure 5-45 shows the energy consumption percentage for all protocols when the number of sources is fifteen and each source sends one packet per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 25.16, 21.80, 21.53, and 16.22 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 20.23, 18.08, 17.19, and 13.29 percent, respectively.

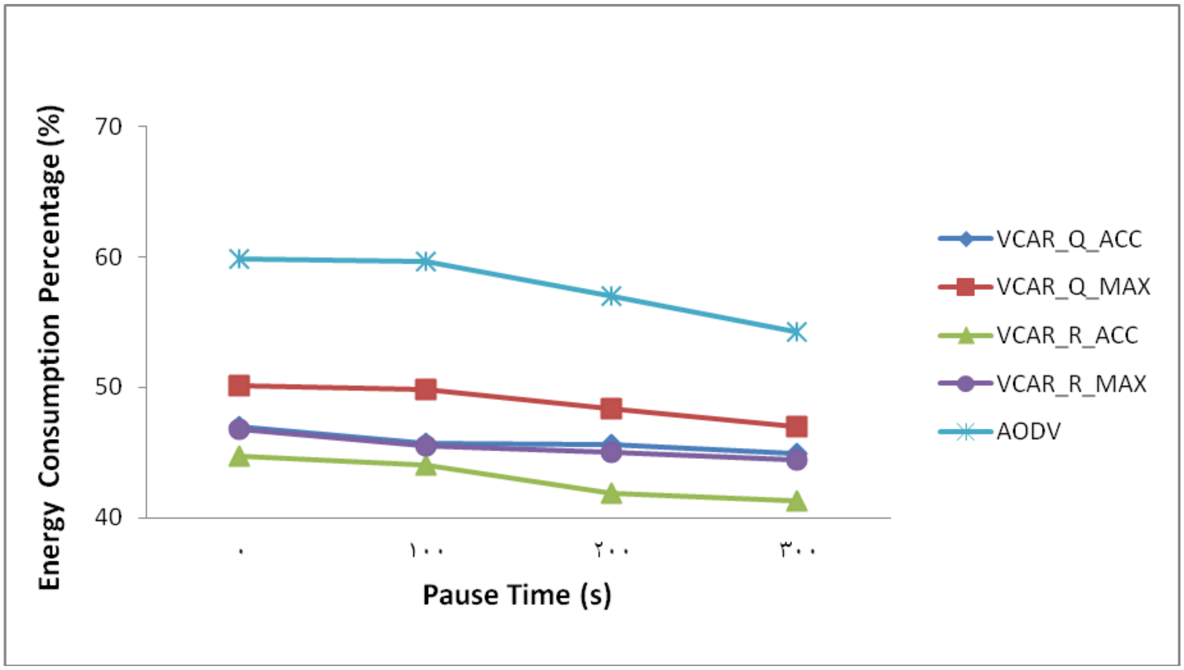


**Figure 5-45: Energy Consumption Percentage of 15 sources each one sends 1 packet/s**

Figure 5-46 shows the energy consumption percentage for all protocols when the number of sources is fifteen and each source sends two packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 20.98, 18.92, 17.67, and 13.46 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 20.36, 17.58, 15.62, and 12.98 percent, respectively.

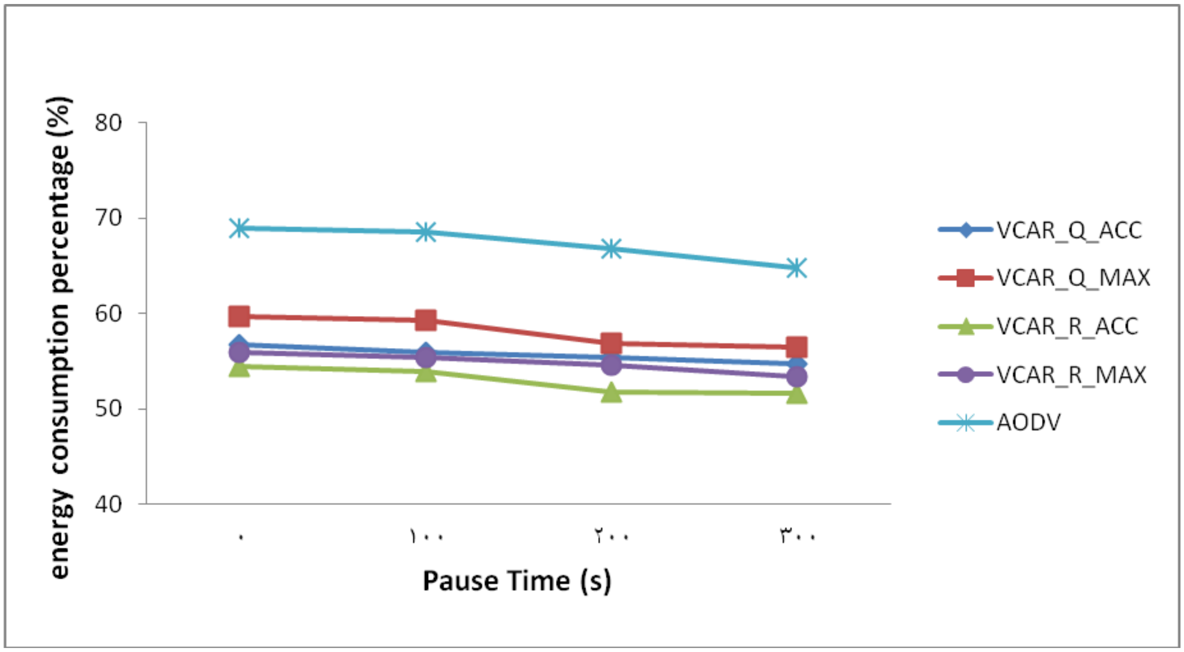


**Figure 5-46:Energy Consumption Percentage of 15 sources each one sends 2 packets/s**

Figure 5-47 shows the energy consumption percentage for all protocols when the number of sources is fifteen and each source sends four packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 17.19, 14.75, 14.02, and 8.70 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 20.17, 16.99, 15.50, and 13.94 percent, respectively.

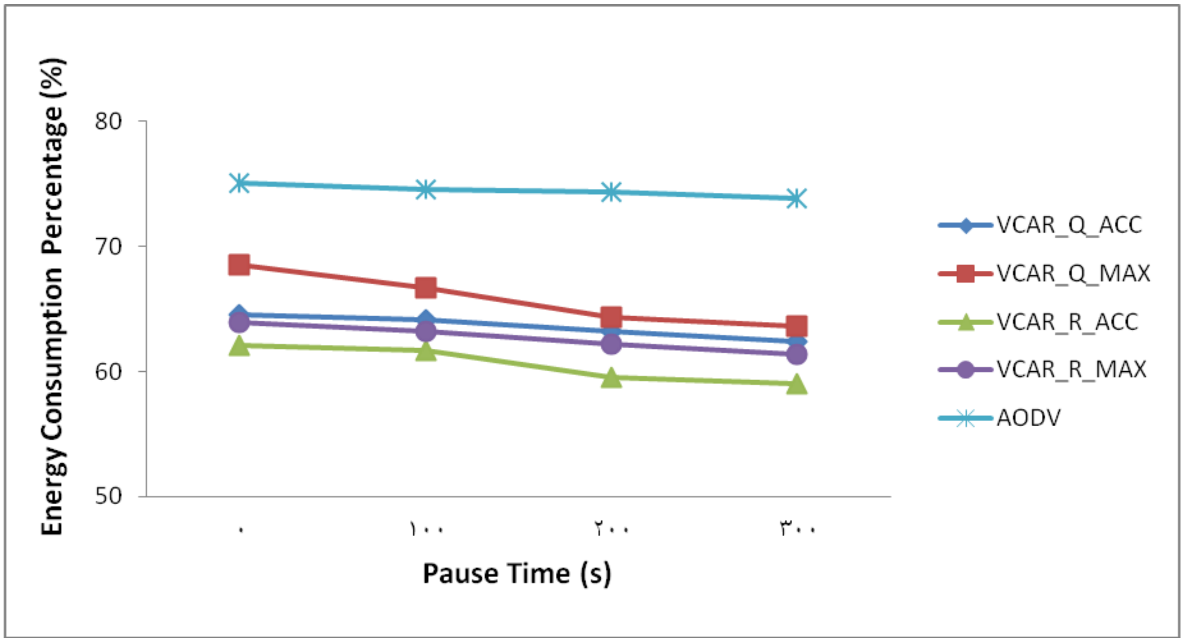


**Figure 5-47:Energy Consumption Percentage of 15 sources each one sends 4 packets/s**

Figure 5-48 shows the energy consumption percentage for all protocols when the number of sources is fifteen and each source sends six packets per second. The figure shows that all VCAR protocols outperform AODV for all pause times. When the pause time is equal to zero (high mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 14.94, 13.44, 11.98, and 10.23 percent, respectively. When the pause time is equal to 300 seconds (low mobility network), VCAR_R_ACC, VCAR_R_MAX, VCAR_Q_ACC, and VCAR_Q_MAX outperform AODV by 17.30, 13.76, 12.71, and 12.20 percent, respectively.

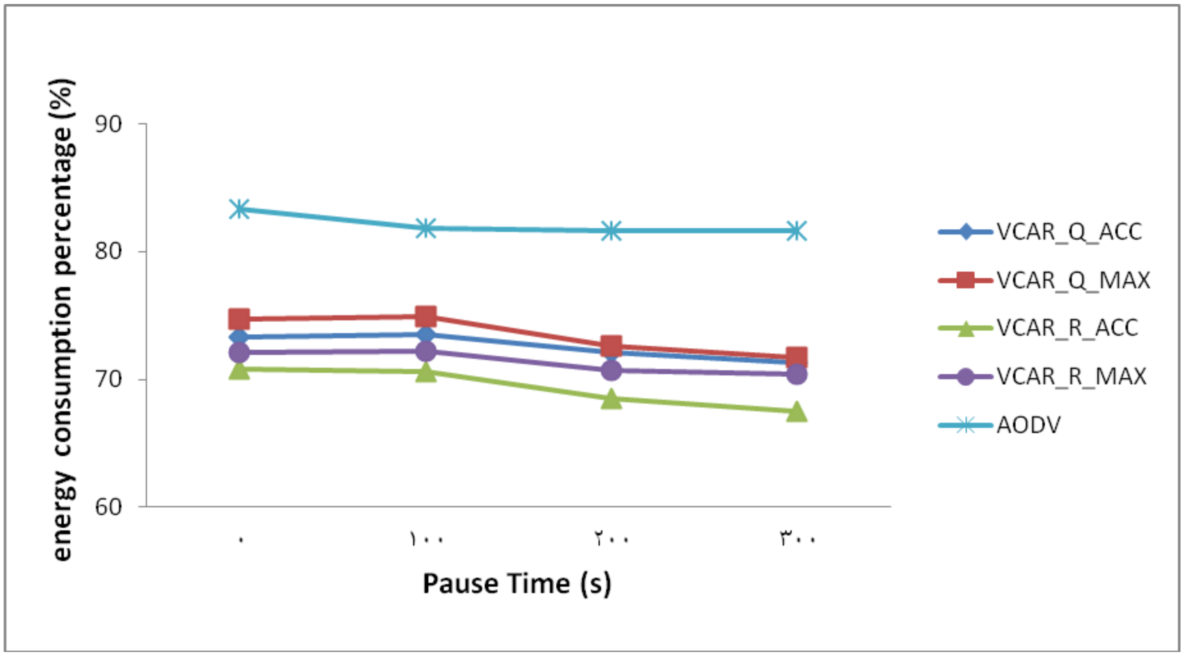


**Figure 5-48:Energy Consumption Percentage of 15 sources each one sends 6 packets/s**

# CHAPTER SIX: CONCLUSIONS AND FUTURE WORK

## 6-1 Conclusions

In this study, we have implemented a new routing protocol for MANETs called VCAR, which selects the route according to both the congestion and the velocity of the intermediate nodes in the network.

VCAR was evaluated against AODV by extensive simulation using NS-2 simulation environment. The performance was evaluated using four metrics: packets delivery ratio, average end-to-end delay, routing overhead, and energy consumption percentage.

We have used different scenarios in the simulation. The nodes move according to the random-waypoint model, with pause times 0, 100, 200, 300 in each experiment. CBR traffic was generated using the transmission rate for 1, 2, 4 and 6 packets per second, repeated for 5, 10, and 15 sources.

The results collected from the simulation show that VCAR outperforms AODV by a significant value in terms of packet delivery ratio, because it selects the more stable route available from source to destination.

VCAR also outperforms AODV by a significant value in terms of average end-to-end delay, because it selects the least congested route from source to destination, so that the packets need not to wait too long in the intermediate nodes.

VCAR also outperforms AODV in terms of energy consumption percentage, because it distributes the load among intermediate nodes, and reduces the energy consumed for route maintenance.

In the other hand, the VCAR routing overhead is higher than the AODV routing overhead, this is because destination in VCAR may sends many route replies for the same route request to select the best route.

## 6-2 Future Work

Future work may include:

- The performance of VCAR may be enhanced by tuning the weight factor (α) dynamically according to the state of the network.
- More studies are needed in the route reply mechanism used in VCAR in order to enhance the routing overhead.

# REFERENCES


1. Bouhorma, M., Bentaouit, H. and Boudhir, A., **Performance Comparison of Ad-hoc Routing Protocols AODV and DSR**, IEEE, International Conference on Multimedia Computing and Systems, October 2009, pp. 511-514.

2. Chen, X., Jones, H. M. and Jayalath, A. D. S., **Congestion-Aware Routing Protocol for Mobile Ad Hoc Networks**, Vehicular Technology Conference 2007, pp. 21-25.

3. Dube, R., Rais, C. D., Wang, K.-Y. and Tripathi, S. K., **Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks**, IEEE Personal Communications Magazine, Vol. 4, No. 1, February 1997, pp. 36–45.

4. Fraser, G., Kevin, C., Jose, S. and Sandra M., **Location and Mobility-Aware Routing for Improving Multimedia Streaming Performance in MANETs**, Wireless Personal Communications, Vol. 83, No. 4, August 2015, pp. 1653-1672.

5. Gupta, A. and Sharma, S. D., **A Survey on Location Based Routing Protocols in Mobile Ad-hoc Networks**, International Journal of Computer Science and Information Technologies, Vol. 5, No. 2, 2014, pp. 994-997.

6. **http://nile.wpi.edu/NS/overview.html**, Access Date: 25/3/2016.

7. Jadeja, N., Patel, R, **"Performance Evaluation of AODV, DSDV and DSR Routing Protocols using NS- 2 Simulator."** International Journal of Engineering Research and Applications (IJERA), Vol. 3, No. 2, pp.1825-1830, Mar. 2013.

8. Jain, S., Kokate, S., Thakur, P. and Takalkar, S., **A Study of Congestion Aware Adaptive Routing Protocols in MANET**, Computer Engineering and Intelligent Systems, Vol. 3, No. 4, 2012, pp. 64-73.

9. Johnson, D. B. and Maltz, D. A., **Dynamic Source Routing in Ad Hoc Wireless Networks**, In Mobile Computing, Kluwer Academic Publishers, Vol. 353,No. 1, 1996, pp. 153-181.

10. Khamayseh, Y., Obiedat, G. and Bani Yassin, M., **Mobility and Load aware Routing protocol for ad hoc networks**, Journal of King Saud University - Computer and Information Sciences, Vol. 23,No. 2, July 2011, pp. 105–113.

11. Khan R., Vishwakarma A., **Enhancement of MANET Routing Protocol.,** IEEE Conference of IT in Business, Industry and Government (CSIBIG), pp. 1-6, 2014.

12. Kavita, P. and Abhishek, S., **A Comprehensive Performance Analysis of Proactive, Reactive and Hybrid MANETs Routing Protocols**, IJCSI International Journal of Computer Science Issues, Vol. 8, No 3, November 2011.

13. Ko, Y.-B. and Vaidya, N. H., **Location-Aided Routing (LAR) in Mobile Ad Hoc Networks**, Wireless Networks Journal, Vol. 6, No. 4, September 2000, pp. 307-321.

14. Lee, S.-J. and Gerla, M., **Dynamic Load-Aware Routing in Ad Hoc Networks**, In Proceeding of the IEEE International Conference of Communication (ICC), Vol. 10, Helinski, Finland, 2001, pp. 3206-3210.

15. Li, X.-Y., **Wireless Ad Hoc and Sensor Networks: Theory and Applications**, Cambridge University Press, 2008.

16. Lochert, C., Scheuermann, B. and Mauve, M., **A Survey on Congestion Control for Mobile Ad-Hoc Networks**, Wiley Wireless Communications and Mobile Computing, Vol. 7, No. 5, June 2007, pp. 655-676.

17. Murthy, S. and Garcia-Luna-Aceves, J. J., **An Efficient Routing Protocol for Wireless Networks**, ACM Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks, Vol. 1, No. 2, October 1996, pp. 183–197.

18. **Object TCL Extensions (OTCL)**. http://bmrc.berkeley.edu/research /cmt/cmtdoc/otcl/, Access Date: 29/1/2016.

19. Park, V. D. and Corson, M. S., **A highly adaptive distributed routing algorithm for mobile wireless networks**, Sixteen annual joint conference of the IEEE computer and communications societies , Vol. 3, April 1997, pp. 1405-1413.

20. Perkins, C. E. and Bhagwat, P., **Highly Dynamic Destination-SequencedDistance-Vector Routing (DSDV) for Mobile Computers**, ACM SIGCOMM Computer Communication Review, Vol. 24, No. 4, October 1994, pp. 234-244.

21. Perkins, C. E. and Royer, E. M., **Ad hoc On-Demand Distance Vector (AODV) Routing**, In Proceedings of the second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99), New Orleans, LA, February 25-26, 1999, pp. 90-100.

22. Rais, K., Ansari, M. N. and Anilkumar, V., **Enhancement of Manet Routing Protocol**, IEEE,2014.

23. Royer, E. M. and Barbara, S., **A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks**, IEEE Personal Communications, April 1999, pp. 46-55.

24. Saurin, J. and Nikhil, N., **A Light-Weight Trust based Mobility Aware Routing Algorithm for Mobile Ad Hoc Networks**, International Journal of Computer Applications, Vol. 97, No. 14, July 2014, pp. 26-30.

25. Seetan, R., Ababneh, I. and Dalal'ah, A., **Congestion-Aware Routing Protocols for Ad Hoc Networks**, The 4th International Conference on Information Technology (ICIT 2009), al-Zaytoona University, Amman, Jordan, June 3rd – 5th, 2009.

26. Shrivastava, L., Tomar, G. S. and Bhadauria, S., **A Survey on Congestion Adaptive Routing Protocols for Mobile Ad-Hoc Networks**, International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011, pp. 189-196.

27. Sweety, G., **Zone Routing Protocol (ZRP) in Ad-Hoc Networks**, International Journal of Research in Engineering and Applied Sciences, Vol. 3, No. 3, March 2013, pp. 92-98.

28. Toh, C.-K., **Ad Hoc Wireless Networks: Protocols and Systems**, Prentice Hall, 2002.

29. Toh, C.-K., **Associativity-Based Routing for Ad Hoc Mobile Networks**, Wireless Personal Communications Journal, Vol. 4, No. 2, March 1997, pp. 103-139.

30. Tran, D. A. and Raghavendra, H., **Congestion Adaptive Routing in Ad Hoc Networks**, IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 11, November 2006, pp. 1294-1305.

31. Yi, Y., Kwon, T. J. and Gerla, M., **A Load aWare Routing (LWR) Based on Local Information**, In Proceedings of the 12th IEEE International Symposium on personal, Indoor and mobile Radio Communications, San Diego, CA, Vol. 2, 2001, pp. 65-69.

## الملخص

الشبكات المتحركة الخاصة هي شبكات لاسلكية تعمل بدون وجود بنية تحتية وتتكون من مجموعة من العُقد المتحركة التي تحتاج إلى بناء مسارات بين بعضها عند الحاجة إلى إرسال البيانات. تستخدم هذه العُقد خوارزميات التمرير لبناء وصيانة المسارات بين المصدر والهدف. إن معظم خوارزميات التمرير المطورة تختار مسارات التمرير دون أن تأخذ بعين الاعتبار كلاً من الازدحام في العُقد الوسطية وحركة هذه العُقد. بعض خوارزميات التمرير التي تم تطويرها تأخذ بعين الاعتبار الازدحام فقط في اختيار المسارات، مما يؤدي إلى اختيار مسارات غير مستقرة مما يزيد من احتمال حدوث أعطال متكررة في المسارات وبالتالي حدوث فقدان متكرر للرزم المرسلة. في حين تعمل خوارزميات أخرى على اختيار المسارات اعتمادا على سرعة العقد الوسطية مما يؤدي إلى اختيار مسارات مزدحمة، وبالتالي حدوث بطء في وصول الرزم من المصدر إلى الهدف.

في هذا البحث تم اقتراح خوارزمية تمرير اعتماداً على السرعة والازدحام في العُقد التي يتكون منها المسار. وتقوم هذه الخوارزمية باختيار المسار الأفضل باستخدام معيارين هما مستوى الازدحام في العُقد الوسطية وسرعة حركة هذه العُقد.

لقياس مستوى الازدحام في العُقد الوسطية تستخدم الخوارزمية المقترحة طريقتين. الطريقة الأولى تتم باستخدام عدد الرزم التي تنتظر في طابور الانتظار في العُقدة والطريقة الثانية تتم باستخدام عدد المسارات التي تمر من خلال العقدة.

تعمل الخوارزمية المقترحة على اختيار المسار الأفضل بأسلوبين. الأسلوب الأول يختار المسار الأفضل بناءاً على القيمة التراكمية لجميع العُقد الوسطية والأسلوب الثاني يختار المسار الأفضل بناءاً على قيمة النهاية العظمى من بين جميع العُقد الوسطية.

تم تنفيذ أربع إصدارات من الخوارزمية المقترحة. الإصدار الأول يعتبر المسار الأفضل هو المسار ذو القيمة التراكمية الأقل من حيث عدد الرزم التي تنتظر في طابور الانتظار وسرعة حركة العُقد الوسطية. الإصدار الثاني يعتبر المسار الأفضل هو المسار ذو القيمة العظمى الأقل من حيث عدد الرزم التي تنتظر في طابور الانتظار وسرعة حركة العُقد الوسطية. الإصدار الثالث يعتبر المسار الأفضل هو المسار ذو القيمة التراكمية الأقل من حيث عدد المسارات التي

تمر من خلال العُقد الوسطية وسرعة حركة العُقد الوسطية. أما الإصدار الرابع فيعتبر المسار الأفضل هو المسار ذو القيمة العظمى الأقل من حيث عدد المسارات التي تمر من خلال العُقد الوسطية وسرعة حركة العُقد الوسطية.

تم إجراء محاكاة مكثفة للخوارزميات المقترحة باستخدام محاكي الشبكة (NS2) وتم استخراج نتائج الخوارزميات المقترحة ومقارنتها مع أداء خوارزمية (AODV) وأظهرت نتائج المحاكاة تحسينات ملحوظة بالنسبة لنسبة تسليم الرزم ومعدل زمن التأخير ونسبة استهلاك الطاقة. ولكن النتائج أظهرت أيضا أن أداء الخوارزميات المقترحة كان أسوأ بالنسبة لمعيار كلفة التمرير.

# APPENDIX A: **RESULTS WHEN α=0.25**

**Table A-1: Packet Delivery Ratio when α=0.25**

| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| sends one | 0 | 86.36 | 85 | 92.49 | 91.25 |
| | 100 | 94.38 | 93.61 | 98.13 | 97.04 |
| packet/second | 200 | 96.2 | 95.74 | 99.12 | 98.23 |
| | 300 | 97.24 | 96.03 | 100 | 100 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends two | 0 | 89.64 | 88.73 | 94.51 | 93.05 |
| | 100 | 94.5 | 94.11 | 97.24 | 96.84 |
| packets/second | 200 | 95.67 | 95.21 | 97.91 | 98.01 |
| | 300 | 96.99 | 96.47 | 100 | 100 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends four | 0 | 90.33 | 89.91 | 91.36 | 90.91 |
| | 100 | 94.22 | 93.42 | 95.84 | 94.35 |
| packets/second | 200 | 93.19 | 92.77 | 97.41 | 95.84 |
| | 300 | 95.48 | 95.19 | 98.13 | 97.55 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends six | 0 | 88.72 | 88.31 | 89.35 | 89.01 |
| | 100 | 90.52 | 89.86 | 91.14 | 90.69 |
| packets/second | 200 | 92.39 | 91.68 | 94.52 | 93.41 |
| | 300 | 95.54 | 95.09 | 97.77 | 96.25 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends one | 0 | 89.05 | 88.74 | 90.28 | 89.91 |
| | 100 | 92.11 | 91.58 | 93.33 | 92.76 |
| packet/second | 200 | 94.54 | 93.83 | 96.36 | 95.81 |
| | 300 | 95.51 | 94.47 | 97.7 | 96.59 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends two | 0 | 89.34 | 89.05 | 90.98 | 90.24 |
| | 100 | 92.87 | 92.33 | 93.91 | 93.25 |
| packets/second | 200 | 94.35 | 93.87 | 96.65 | 95.42 |
| | 300 | 95.48 | 94.13 | 97.05 | 96.48 |
| | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Ten sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 90.95 | 90.27 | 91.54 | 91.05 |
| | 100 | 81.84 | 81.29 | 82.94 | 82.15 |
| | 200 | 93.88 | 93.62 | 95.98 | 95.64 |
| | 300 | 95.62 | 94.41 | 97.23 | 96.87 |
| | | | | | |
| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 83.62 | 83.54 | 84.16 | 84.08 |
| | 100 | 67.62 | 64.84 | 69.25 | 68.11 |
| | 200 | 72.12 | 71.91 | 73.84 | 72.65 |
| | 300 | 89.91 | 89.17 | 91.52 | 90.51 |
| | | | | | |
| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 90.92 | 90.55 | 91.55 | 91.45 |
| | 100 | 94.13 | 93.98 | 95.88 | 95.14 |
| | 200 | 95.82 | 95.15 | 97.16 | 96.74 |
| | 300 | 98.14 | 98.04 | 98.91 | 98.85 |
| | | | | | |
| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 89.49 | 88.66 | 91.02 | 90.54 |
| | 100 | 91.51 | 90.16 | 93.67 | 92.81 |
| | 200 | 93.34 | 92.05 | 95.23 | 94.11 |
| | 300 | 94.43 | 94.12 | 96.91 | 95.74 |
| | | | | | |
| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 78.56 | 77.22 | 80.65 | 79.12 |
| | 100 | 68.12 | 66.94 | 70.05 | 69.51 |
| | 200 | 74.35 | 74.05 | 76.12 | 75.3 |
| | 300 | 81.95 | 81.55 | 82.61 | 82.05 |
| | | | | | |
| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 61.51 | 61.08 | 63.02 | 62.84 |
| | 100 | 50.13 | 49.85 | 51.81 | 50.44 |
| | 200 | 55.64 | 54.71 | 57.14 | 56.41 |
| | 300 | 64.02 | 63.77 | 65.31 | 64.51 |

**Table A-2: Average End-to-End Delay when α=0.25**

| | | | | | |
|---|---|---|---|---|---|
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends one | 0 | 0.4178 | 0.4419 | 0.3244 | 0.3502 |
| | 100 | 0.3114 | 0.3952 | 0.2418 | 0.2711 |
| packet/second | 200 | 0.0667 | 0.0702 | 0.0502 | 0.0592 |
| | 300 | 0.0228 | 0.0294 | 0.0134 | 0.0203 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends two | 0 | 0.1728 | 0.1763 | 0.1662 | 0.1705 |
| | 100 | 0.1864 | 0.1926 | 0.1601 | 0.1788 |
| packets/second | 200 | 0.085 | 0.0912 | 0.0642 | 0.0778 |
| | 300 | 0.0239 | 0.0252 | 0.0197 | 0.0215 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends four | 0 | 0.3601 | 0.3696 | 0.3455 | 0.3549 |
| | 100 | 0.1744 | 0.1752 | 0.1502 | 0.1629 |
| packets/second | 200 | 0.0602 | 0.0667 | 0.0512 | 0.0561 |
| | 300 | 0.0274 | 0.0281 | 0.0212 | 0.0229 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends six | 0 | 0.2025 | 0.2305 | 0.1692 | 0.1837 |
| | 100 | 0.2448 | 0.249 | 0.2194 | 0.2208 |
| packets/second | 200 | 0.0639 | 0.0651 | 0.0606 | 0.0624 |
| | 300 | 0.0274 | 0.0281 | 0.0211 | 0.0262 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends one | 0 | 0.1779 | 0.1806 | 0.1662 | 0.1724 |
| | 100 | 0.2405 | 0.2487 | 0.2344 | 0.2391 |
| packet/second | 200 | 0.0602 | 0.0619 | 0.0584 | 0.0591 |
| | 300 | 0.0306 | 0.0311 | 0.0282 | 0.0294 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends two | 0 | 0.0971 | 0.0986 | 0.0914 | 0.0952 |
| | 100 | 0.2091 | 0.2112 | 0.1968 | 0.2007 |
| packets/second | 200 | 0.0514 | 0.0526 | 0.0488 | 0.0509 |
| | 300 | 0.0423 | 0.0435 | 0.0394 | 0.041 |
| | | | | | |

| Ten sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.1369 | 0.1394 | 0.1305 | 0.1334 |
| | 100 | 0.3611 | 0.3678 | 0.3507 | 0.3594 |
| | 200 | 0.0904 | 0.0913 | 0.0891 | 0.0899 |
| | 300 | 0.1368 | 0.1397 | 0.1256 | 0.1311 |

| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.1994 | 0.2016 | 0.1928 | 0.1957 |
| | 100 | 0.7711 | 0.7809 | 0.7501 | 0.7607 |
| | 200 | 0.4601 | 0.4715 | 0.4109 | 0.4315 |
| | 300 | 0.2218 | 0.2307 | 0.2005 | 0.2109 |

| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.1102 | 0.1118 | 0.1048 | 0.1094 |
| | 100 | 0.1347 | 0.1376 | 0.1314 | 0.1335 |
| | 200 | 0.0552 | 0.0571 | 0.0512 | 0.0534 |
| | 300 | 0.0468 | 0.0491 | 0.0448 | 0.0455 |

| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.0922 | 0.0934 | 0.0905 | 0.0912 |
| | 100 | 0.1638 | 0.1645 | 0.1611 | 0.1627 |
| | 200 | 0.0579 | 0.0582 | 0.0554 | 0.0568 |
| | 300 | 0.0519 | 0.0526 | 0.0492 | 0.0504 |

| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.2335 | 0.2391 | 0.2294 | 0.2314 |
| | 100 | 0.6208 | 0.6233 | 0.6159 | 0.6165 |
| | 200 | 0.5218 | 0.5266 | 0.5088 | 0.5142 |
| | 300 | 0.4508 | 0.4552 | 0.4415 | 0.4463 |

| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.4483 | 0.4491 | 0.4426 | 0.4468 |
| | 100 | 1.1308 | 1.1416 | 1.1056 | 1.1125 |
| | 200 | 1.2271 | 1.2391 | 1.2041 | 1.2151 |
| | 300 | 1.1294 | 1.1402 | 1.1042 | 1.1119 |

**Table A-3: Routing Overhead when α=0.25**

| | | | | | |
|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 2.95 | 3.08 | 2.64 | 2.77 |
| | 100 | 2.47 | 2.84 | 2.35 | 2.51 |
| | 200 | 2.42 | 2.66 | 2.04 | 2.29 |
| | 300 | 1.88 | 2.09 | 1.67 | 1.51 |
| | | | | | |
| Five sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 1.95 | 1.99 | 1.84 | 1.91 |
| | 100 | 1.61 | 1.69 | 1.47 | 1.55 |
| | 200 | 1.42 | 1.46 | 1.31 | 1.35 |
| | 300 | 1.34 | 1.42 | 1.15 | 1.27 |
| | | | | | |
| Five sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 4.12 | 4.33 | 3.42 | 3.91 |
| | 100 | 2.93 | 2.98 | 2.74 | 2.89 |
| | 200 | 3.04 | 3.19 | 2.56 | 2.87 |
| | 300 | 2.19 | 2.25 | 2.07 | 2.12 |
| | | | | | |
| Five sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 3.56 | 3.67 | 3.33 | 3.41 |
| | 100 | 2.73 | 2.91 | 2.51 | 2.69 |
| | 200 | 2.09 | 2.24 | 1.74 | 1.9 |
| | 300 | 1.85 | 1.93 | 1.41 | 1.62 |
| | | | | | |
| Ten sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 3.81 | 3.88 | 3.66 | 3.71 |
| | 100 | 3.15 | 3.31 | 2.97 | 3.04 |
| | 200 | 2.5 | 2.68 | 2.35 | 2.41 |
| | 300 | 2.13 | 2.19 | 1.91 | 2.05 |
| | | | | | |
| Ten sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 2.89 | 2.94 | 2.77 | 2.84 |
| | 100 | 2.81 | 2.97 | 2.68 | 2.74 |
| | 200 | 1.99 | 2.01 | 1.94 | 1.97 |
| | 300 | 1.83 | 1.94 | 1.58 | 1.74 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |

| | | | | | |
|---|---|---|---|---|---|
| sends four packets/second | 0 | 2.88 | 2.93 | 2.75 | 2.81 |
| | 100 | 2.76 | 2.84 | 2.54 | 2.63 |
| | 200 | 2.04 | 2.15 | 1.91 | 1.99 |
| | 300 | 1.84 | 1.95 | 1.62 | 1.77 |
| | | | | | |
| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 2.69 | 2.74 | 2.48 | 2.56 |
| | 100 | 2.25 | 2.34 | 2.04 | 2.15 |
| | 200 | 1.95 | 2.04 | 1.7 | 1.85 |
| | 300 | 1.58 | 1.64 | 1.31 | 1.49 |
| | | | | | |
| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 4.05 | 4.12 | 3.82 | 3.91 |
| | 100 | 3.71 | 3.92 | 3.52 | 3.64 |
| | 200 | 3.27 | 3.35 | 3.09 | 3.16 |
| | 300 | 3.14 | 3.21 | 2.99 | 3.07 |
| | | | | | |
| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 3.05 | 3.13 | 2.81 | 2.92 |
| | 100 | 2.63 | 2.66 | 2.54 | 2.59 |
| | 200 | 2.06 | 2.14 | 1.92 | 1.99 |
| | 300 | 1.88 | 1.95 | 1.52 | 1.67 |
| | | | | | |
| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 2.58 | 2.64 | 2.33 | 2.41 |
| | 100 | 2.11 | 2.25 | 1.85 | 1.99 |
| | 200 | 1.56 | 1.69 | 1.35 | 1.47 |
| | 300 | 1.62 | 1.84 | 1.15 | 1.24 |
| | | | | | |
| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 1.86 | 1.94 | 1.55 | 1.67 |
| | 100 | 1.57 | 1.6 | 1.34 | 1.42 |
| | 200 | 1.36 | 1.49 | 1.15 | 1.24 |
| | 300 | 1.56 | 1.62 | 1.31 | 1.47 |

**Table A-4: Energy Consumption Percentage when α=0.25**

| | | | | | |
|---|---|---|---|---|---|
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends one | 0 | 22.6214 | 23.0527 | 22.0112 | 22.2511 |
| | 100 | 24.5188 | 25.5611 | 22.1924 | 22.145 |
| packet/second | 200 | 20.1025 | 21.281 | 19.7225 | 19.9499 |
| | 300 | 19.9447 | 20.9049 | 18.3255 | 19.1496 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends two | 0 | 24.5199 | 25.1096 | 23.4136 | 23.9048 |
| | 100 | 27.0051 | 29.3681 | 25.441 | 26.7513 |
| packets/second | 200 | 24.6625 | 25.2276 | 23.9354 | 24.8152 |
| | 300 | 23.4082 | 24.9157 | 22.1289 | 24.2099 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends four | 0 | 44.6914 | 46.0105 | 39.7124 | 41.5823 |
| | 100 | 47.3815 | 49.9127 | 42.1189 | 45.1806 |
| packets/second | 200 | 38.6944 | 38.107 | 36.2881 | 37.4166 |
| | 300 | 35.6488 | 36.1492 | 34.1264 | 34.9005 |
| | | | | | |
| Five sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends six | 0 | 57.3631 | 59.0058 | 53.1684 | 55.9871 |
| | 100 | 62.6712 | 64.1823 | 57.9118 | 60.0048 |
| packets/second | 200 | 51.3514 | 51.9803 | 50.1182 | 50.9488 |
| | 300 | 49.1825 | 49.8522 | 44.9258 | 47.3522 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends one | 0 | 31.0558 | 31.9024 | 28.1425 | 30.4418 |
| | 100 | 35.1863 | 36.2088 | 33.1204 | 34.0255 |
| packet/second | 200 | 31.3947 | 32.1558 | 28.3035 | 30.9044 |
| | 300 | 29.9076 | 30.1184 | 26.4712 | 29.115 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| sends two | 0 | 42.0778 | 43.1725 | 39.2114 | 40.8256 |
| | 100 | 48.1185 | 49.4618 | 46.2914 | 47.5182 |
| packets/second | 200 | 42.0582 | 42.8273 | 40.9112 | 41.5722 |
| | 300 | 36.7582 | 37.7522 | 36.312 | 37.8436 |
| | | | | | |
| Ten sources each one | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |

| | | | | | |
|---|---|---|---|---|---|
| sends four packets/second | 0 | 61.6184 | 62.8724 | 59.2841 | 60.357 |
| | 100 | 76.4429 | 76.9831 | 73.0059 | 74.5896 |
| | 200 | 68.0593 | 69.9211 | 65.1479 | 66.6971 |
| | 300 | 60.3641 | 61.2776 | 58.0022 | 59.4682 |
| | | | | | |
| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 82.5978 | 83.0478 | 79.2584 | 80.3647 |
| | 100 | 84.1814 | 84.7691 | 83.4935 | 83.978 |
| | 200 | 84.9823 | 85.306 | 82.2831 | 83.9921 |
| | 300 | 73.9498 | 74.7055 | 72.1628 | 72.8671 |
| | | | | | |
| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 35.7204 | 36.5478 | 33.5198 | 34.2487 |
| | 100 | 38.5611 | 38.9155 | 36.8221 | 37.0489 |
| | 200 | 38.5697 | 39.0047 | 33.4189 | 37.2941 |
| | 300 | 32.3778 | 34.8256 | 30.3051 | 31.8452 |
| | | | | | |
| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 51.9544 | 52.0047 | 50.1584 | 51.3641 |
| | 100 | 57.3814 | 58.6284 | 55.8513 | 56.1482 |
| | 200 | 52.5825 | 53.3716 | 50.8521 | 51.4497 |
| | 300 | 50.6721 | 51.0852 | 48.3252 | 49.0026 |
| | | | | | |
| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 82.2792 | 83.1047 | 80.4459 | 81.367 |
| | 100 | 83.6472 | 84.4426 | 81.5712 | 82.9147 |
| | 200 | 82.9258 | 83.5528 | 80.2591 | 81.0159 |
| | 300 | 76.508 | 77.1184 | 74.6462 | 75.4738 |
| | | | | | |
| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 89.0059 | 89.1578 | 88.0157 | 88.6347 |
| | 100 | 88.9528 | 89.4426 | 87.9501 | 88.1056 |
| | 200 | 86.9561 | 87.2914 | 85.9411 | 86.047 |
| | 300 | 81.1834 | 81.7623 | 80.0451 | 80.6834 |

# APPENDIX B: RESULTS WHEN α=0.75

**Table B-1: Packet Delivery Ratio when α=0.75**

| Five sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 90.32 | 89.98 | 92.25 | 91.08 |
| | 100 | 97.02 | 96.81 | 98.13 | 97.52 |
| | 200 | 98.05 | 97.91 | 99.25 | 98.94 |
| | 300 | 100 | 100 | 100 | 100 |
| | | | | | |
| Five sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 92.15 | 91.95 | 94.25 | 93.64 |
| | 100 | 96.95 | 96.66 | 97.85 | 97.05 |
| | 200 | 98.06 | 97.82 | 99.61 | 98.82 |
| | 300 | 100 | 100 | 100 | 100 |
| | | | | | |
| Five sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 93.25 | 92.94 | 95.82 | 95.12 |
| | 100 | 97.52 | 96.91 | 98.85 | 98.63 |
| | 200 | 98.15 | 97.56 | 98.99 | 98.62 |
| | 300 | 100 | 100 | 100 | 100 |
| | | | | | |
| Five sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 94.13 | 93.68 | 95.44 | 94.95 |
| | 100 | 95.84 | 95.11 | 96.91 | 96.06 |
| | 200 | 97.82 | 97.75 | 98.95 | 98.15 |
| | 300 | 100 | 100 | 100 | 100 |
| | | | | | |
| Ten sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 93.81 | 93.11 | 94.95 | 94.05 |
| | 100 | 96.11 | 95.96 | 97.62 | 96.98 |
| | 200 | 97.51 | 97.32 | 98.85 | 98.05 |
| | 300 | 100 | 100 | 100 | 100 |
| | | | | | |
| Ten sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 93.12 | 92.84 | 94.63 | 93.91 |
| | 100 | 95.85 | 95.11 | 97.41 | 96.92 |
| | 200 | 98.95 | 98.81 | 99.95 | 99.12 |
| | 300 | 100 | 100 | 100 | 100 |
| | | | | | |

| Ten sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 93.54 | 92.98 | 96.24 | 94.12 |
| | 100 | 85.36 | 84.81 | 87.2 | 86.15 |
| | 200 | 97.81 | 97.42 | 98.56 | 98.05 |
| | 300 | 99.21 | 99.09 | 99.56 | 99.44 |

| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 89.54 | 88.09 | 91.62 | 90.15 |
| | 100 | 73.88 | 72.91 | 75.92 | 74.25 |
| | 200 | 75.55 | 74.98 | 77.63 | 76.53 |
| | 300 | 94.06 | 93.53 | 96.42 | 95.15 |

| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 94.14 | 93.62 | 97.41 | 96.62 |
| | 100 | 97.42 | 96.98 | 98.04 | 97.62 |
| | 200 | 98.75 | 98.65 | 99.52 | 99.11 |
| | 300 | 100 | 100 | 100 | 100 |

| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 94.51 | 94.66 | 96.47 | 95.74 |
| | 100 | 95.99 | 95.71 | 97.88 | 96.95 |
| | 200 | 98.42 | 98.05 | 98.95 | 98.66 |
| | 300 | 100 | 100 | 100 | 100 |

| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 84.35 | 83.33 | 86.35 | 84.62 |
| | 100 | 73.15 | 72.14 | 75.82 | 74.09 |
| | 200 | 81.26 | 80.22 | 82.19 | 81.85 |
| | 300 | 84.16 | 83.55 | 87.15 | 85.84 |

| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 67.15 | 66.6 | 70.26 | 68.25 |
| | 100 | 55.09 | 54.27 | 57.41 | 56.33 |
| | 200 | 61.72 | 60.79 | 63.19 | 62.94 |
| | 300 | 68.52 | 67.53 | 70.14 | 69.52 |

**Table B-2: Average End-to-End Delay when α=0.75**

| | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| Five sources each one sends one packet/second | 0 | 0.3101 | 0.3209 | 0.3012 | 0.3094 |
| | 100 | 0.2391 | 0.2451 | 0.2214 | 0.2359 |
| | 200 | 0.0523 | 0.0542 | 0.0499 | 0.0511 |
| | 300 | 0.014 | 0.0142 | 0.0133 | 0.0138 |
| | | | | | |
| Five sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 0.1423 | 0.1475 | 0.1341 | 0.1391 |
| | 100 | 0.1392 | 0.1412 | 0.1306 | 0.1371 |
| | 200 | 0.0522 | 0.0541 | 0.0491 | 0.0509 |
| | 300 | 0.0146 | 0.015 | 0.0133 | 0.0142 |
| | | | | | |
| Five sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 0.2911 | 0.3056 | 0.2715 | 0.2803 |
| | 100 | 0.0915 | 0.0923 | 0.0808 | 0.0866 |
| | 200 | 0.0352 | 0.0364 | 0.0322 | 0.0334 |
| | 300 | 0.0149 | 0.0153 | 0.0134 | 0.0145 |
| | | | | | |
| Five sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 0.1405 | 0.1423 | 0.1322 | 0.1362 |
| | 100 | 0.2011 | 0.2025 | 0.1925 | 0.1995 |
| | 200 | 0.0536 | 0.0551 | 0.0512 | 0.0524 |
| | 300 | 0.0155 | 0.0159 | 0.0149 | 0.0151 |
| | | | | | |
| Ten sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 0.1472 | 0.151 | 0.1391 | 0.1421 |
| | 100 | 0.2107 | 0.2116 | 0.2022 | 0.2094 |
| | 200 | 0.0523 | 0.0544 | 0.0492 | 0.0509 |
| | 300 | 0.0239 | 0.0245 | 0.0221 | 0.0235 |
| | | | | | |
| Ten sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 0.0795 | 0.0806 | 0.0766 | 0.0782 |
| | 100 | 0.1823 | 0.1882 | 0.1711 | 0.1762 |
| | 200 | 0.0425 | 0.0429 | 0.0392 | 0.0411 |
| | 300 | 0.0345 | 0.0351 | 0.0325 | 0.0335 |
| | | | | | |

| Ten sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.1205 | 0.1255 | 0.1145 | 0.1192 |
| | 100 | 0.2715 | 0.2805 | 0.262 | 0.2688 |
| | 200 | 0.0811 | 0.0852 | 0.0772 | 0.0798 |
| | 300 | 0.1342 | 0.1367 | 0.1255 | 0.1312 |

| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.1719 | 0.1786 | 0.1625 | 0.1675 |
| | 100 | 0.641 | 0.6558 | 0.605 | 0.6229 |
| | 200 | 0.4493 | 0.4527 | 0.4262 | 0.4381 |
| | 300 | 0.2315 | 0.2388 | 0.2155 | 0.2205 |

| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.0942 | 0.0962 | 0.0885 | 0.0911 |
| | 100 | 0.114 | 0.118 | 0.107 | 0.111 |
| | 200 | 0.0461 | 0.0473 | 0.0435 | 0.0452 |
| | 300 | 0.0321 | 0.0341 | 0.0305 | 0.0315 |

| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.0761 | 0.0782 | 0.0741 | 0.0752 |
| | 100 | 0.1512 | 0.1523 | 0.1442 | 0.1492 |
| | 200 | 0.0459 | 0.0471 | 0.0431 | 0.0445 |
| | 300 | 0.0425 | 0.0443 | 0.0394 | 0.0412 |

| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.2113 | 0.2151 | 0.2012 | 0.2071 |
| | 100 | 0.5617 | 0.572 | 0.5514 | 0.5534 |
| | 200 | 0.4512 | 0.4579 | 0.4382 | 0.4476 |
| | 300 | 0.4052 | 0.4192 | 0.3801 | 0.3941 |

| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 0.4182 | 0.42005 | 0.3951 | 0.4021 |
| | 100 | 0.9971 | 1.005 | 0.9541 | 0.9735 |
| | 200 | 1.0995 | 1.1009 | 1.0251 | 1.0741 |
| | 300 | 0.9811 | 0.9905 | 0.9422 | 0.9624 |

**Table B-3: Routing Overhead when α=0.75**

| Five sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 2.86 | 2.99 | 2.45 | 2.79 |
| | 100 | 2.16 | 2.29 | 1.98 | 2.05 |
| | 200 | 2.09 | 2.15 | 1.99 | 2.03 |
| | 300 | 1.69 | 1.9 | 1.47 | 1.56 |
| | | | | | |
| Five sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 1.95 | 1.99 | 1.85 | 1.9 |
| | 100 | 1.56 | 1.6 | 1.47 | 1.5 |
| | 200 | 1.39 | 1.45 | 1.27 | 1.31 |
| | 300 | 1.26 | 1.3 | 1.09 | 1.19 |
| | | | | | |
| Five sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 3.37 | 3.42 | 3.29 | 3.31 |
| | 100 | 2.86 | 2.9 | 2.71 | 2.79 |
| | 200 | 2.7 | 2.74 | 2.49 | 2.61 |
| | 300 | 2.27 | 2.36 | 1.99 | 2.09 |
| | | | | | |
| Five sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 3.36 | 3.49 | 3.08 | 3.19 |
| | 100 | 2.57 | 2.6 | 2.39 | 2.49 |
| | 200 | 1.79 | 1.83 | 1.69 | 1.72 |
| | 300 | 1.42 | 1.49 | 1.32 | 1.36 |
| | | | | | |
| Ten sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 3.8 | 3.96 | 3.57 | 3.69 |
| | 100 | 3.17 | 3.3 | 2.99 | 3.05 |
| | 200 | 2.48 | 2.56 | 2.29 | 2.39 |
| | 300 | 1.96 | 2.01 | 1.79 | 1.86 |
| | | | | | |
| Ten sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 2.79 | 2.8 | 2.69 | 2.75 |
| | 100 | 2.68 | 2.71 | 2.56 | 2.63 |
| | 200 | 1.96 | 2.04 | 1.86 | 1.91 |
| | 300 | 1.8 | 1.85 | 1.6 | 1.67 |
| | | | | | |

| Ten sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 2.75 | 2.79 | 2.68 | 2.71 |
| | 100 | 2.59 | 2.63 | 2.51 | 2.56 |
| | 200 | 1.98 | 2.02 | 1.88 | 1.93 |
| | 300 | 1.7 | 1.74 | 1.6 | 1.64 |

| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 2.61 | 2.67 | 2.37 | 2.52 |
| | 100 | 2.28 | 2.33 | 2.03 | 2.09 |
| | 200 | 1.58 | 1.6 | 1.46 | 1.51 |
| | 300 | 1.37 | 1.41 | 1.25 | 1.31 |

| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 3.96 | 4.02 | 3.69 | 3.84 |
| | 100 | 3.59 | 3.68 | 3.39 | 3.51 |
| | 200 | 3.19 | 3.31 | 3.05 | 3.09 |
| | 300 | 3.02 | 3.11 | 2.89 | 2.94 |

| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 2.88 | 2.91 | 2.77 | 2.81 |
| | 100 | 2.58 | 2.6 | 2.44 | 2.53 |
| | 200 | 1.9 | 1.95 | 1.79 | 1.84 |
| | 300 | 1.49 | 1.52 | 1.42 | 1.47 |

| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 2.37 | 2.39 | 2.28 | 2.31 |
| | 100 | 1.92 | 1.95 | 1.81 | 1.86 |
| | 200 | 1.39 | 1.42 | 1.31 | 1.37 |
| | 300 | 1.2 | 1.25 | 1.09 | 1.14 |

| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 1.56 | 1.65 | 1.47 | 1.51 |
| | 100 | 1.34 | 1.41 | 1.26 | 1.29 |
| | 200 | 1.23 | 1.26 | 1.1 | 1.19 |
| | 300 | 1.2 | 1.24 | 1.09 | 1.16 |

**Table B-4: Energy Consumption Percentage when α=0.75**

| | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 20.8423 | 21.0062 | 19.2571 | 20.2051 |
| | 100 | 20.9145 | 21.6402 | 19.5247 | 20.5061 |
| | 200 | 18.8531 | 19.2012 | 17.3145 | 18.251 |
| | 300 | 17.2105 | 17.7533 | 16.1682 | 16.9875 |
| | | | | | |
| Five sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 21.1358 | 21.9251 | 20.5815 | 20.9487 |
| | 100 | 22.8257 | 23.1594 | 21.9825 | 22.1056 |
| | 200 | 20.4253 | 21.5591 | 19.1982 | 19.9823 |
| | 300 | 20.4235 | 20.9915 | 18.5249 | 19.1503 |
| | | | | | |
| Five sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 35.0056 | 36.4891 | 32.651 | 34.9213 |
| | 100 | 37.1594 | 37.9926 | 35.1258 | 36.6482 |
| | 200 | 33.9158 | 34.0052 | 30.3054 | 32.1159 |
| | 300 | 30.3052 | 31.8251 | 27.5681 | 29.0062 |
| | | | | | |
| Five sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 42.6381 | 43.4369 | 40.1592 | 41.5921 |
| | 100 | 49.5266 | 51.5126 | 46.3105 | 48.5261 |
| | 200 | 41.4925 | 42.5812 | 38.1256 | 39.2251 |
| | 300 | 37.8625 | 39.0062 | 34.3526 | 36.5261 |
| | | | | | |
| Ten sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 24.5923 | 25.7435 | 22.4512 | 23.6215 |
| | 100 | 26.6041 | 26.9005 | 24.5162 | 25.9871 |
| | 200 | 25.2581 | 26.0059 | 22.8403 | 24.5162 |
| | 300 | 23.6591 | 24.2489 | 21.5941 | 22.6931 |
| | | | | | |
| Ten sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 34.0059 | 35.1189 | 31.625 | 33.4259 |
| | 100 | 39.3901 | 41.5582 | 37.4982 | 38.4921 |
| | 200 | 36.2259 | 38.0047 | 33.4925 | 35.4469 |
| | 300 | 34.9562 | 35.0048 | 31.5271 | 33.456 |
| | | | | | |

| Ten sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
|---|---|---|---|---|---|
| | 0 | 54.2591 | 55.3698 | 50.5741 | 52.3655 |
| | 100 | 67.2515 | 69.0014 | 63.0059 | 65.2689 |
| | 200 | 59.2681 | 61.5294 | 55.2256 | 57.261 |
| | 300 | 54.2192 | 54.9821 | 52.2561 | 53.3156 |
| | | | | | |
| Ten sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 73.6259 | 75.0045 | 70.4256 | 72.1519 |
| | 100 | 74.6982 | 75.4135 | 72.1956 | 73.6984 |
| | 200 | 74.2351 | 75.2142 | 71.9852 | 73.4025 |
| | 300 | 65.2081 | 66.6288 | 62.524 | 64.5291 |
| | | | | | |
| Fifteen sources each one sends one packet/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 31.9158 | 32.0058 | 29.1315 | 31.1584 |
| | 100 | 32.6175 | 33.4911 | 30.2193 | 31.5802 |
| | 200 | 30.3052 | 31.9806 | 28.014 | 29.3614 |
| | 300 | 28.5611 | 28.9628 | 26.2734 | 27.881 |
| | | | | | |
| Fifteen sources each one sends two packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 46.2294 | 47.0058 | 42.3632 | 44.9152 |
| | 100 | 49.1058 | 50.4822 | 47.5274 | 48.1266 |
| | 200 | 43.805 | 44.1142 | 41.4109 | 42.2825 |
| | 300 | 42.0052 | 44.4809 | 40.5281 | 41.8536 |
| | | | | | |
| Fifteen sources each one sends four packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 73.6822 | 75.118 | 70.2595 | 72.3569 |
| | 100 | 74.9184 | 76.0051 | 71.5591 | 73.3056 |
| | 200 | 73.8824 | 76.1542 | 70.3521 | 72.1718 |
| | 300 | 71.9522 | 72.5496 | 69.1482 | 71.0082 |
| | | | | | |
| Fifteen sources each one sends six packets/second | Pause Time | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX |
| | 0 | 81.4592 | 82.5561 | 78.4159 | 79.2045 |
| | 100 | 82.5694 | 83.5529 | 78.1215 | 80.806 |
| | 200 | 78.6173 | 79.8259 | 76.6482 | 77.9205 |
| | 300 | 74.5917 | 75.2845 | 72.2458 | 73.3589 |

# APPENDIX C: 95% CONFIDENCE IINTERVAL OF PACKET DELIVERY RATION WHEN α=0.50

**Table C-1: Confidence Interval of Packet Delivery Ratio with Pause Time=0**

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 88.157 | 85.427 | 96.040 | 91.616 | 84.131 |
| | Max | 89.237 | 86.800 | 96.658 | 93.191 | 84.990 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 84.631 | 82.010 | 92.198 | 87.952 | 79.924 |
| | Max | 85.667 | 83.328 | 92.792 | 89.464 | 80.741 |
| Five sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 80.399 | 77.909 | 87.589 | 83.554 | 75.808 |
| | Max | 81.384 | 79.162 | 88.152 | 84.991 | 76.583 |
| Five sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 75.977 | 73.624 | 82.771 | 78.958 | 71.903 |
| | Max | 76.907 | 74.807 | 83.304 | 80.316 | 72.638 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Ten sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 87.284 | 84.581 | 95.089 | 90.709 | 82.724 |
| | Max | 88.353 | 85.941 | 95.701 | 92.269 | 83.570 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 86.079 | 79.468 | 93.776 | 89.457 | 76.425 |
| | Max | 87.133 | 80.810 | 94.380 | 90.995 | 77.256 |
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 76.378 | 68.541 | 85.713 | 79.323 | 66.656 |
| | Max | 77.362 | 69.827 | 86.267 | 80.785 | 67.432 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 68.929 | 60.366 | 75.660 | 69.296 | 57.444 |
| | Max | 69.774 | 61.499 | 76.127 | 70.574 | 58.112 |
| Fifteen sources each one sends | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| one packets/second | Min | 85.736 | 79.152 | 93.403 | 89.100 | 76.166 |
| | Max | 86.786 | 80.488 | 94.004 | 90.632 | 76.994 |
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 75.350 | 65.277 | 85.441 | 75.546 | 63.725 |
| | Max | 76.288 | 66.502 | 85.968 | 76.938 | 64.467 |
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 68.348 | 59.887 | 74.948 | 68.678 | 57.101 |
| | Max | 69.186 | 61.011 | 75.411 | 69.944 | 57.766 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 61.025 | 54.942 | 65.744 | 62.434 | 51.628 |
| | Max | 61.77 | 55.973 | 66.149 | 63.585 | 52.229 |

**Table C-2: Confidence Interval of Packet Delivery Ratio with Pause Time=300**

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 98.001 | 95.331 | 99.946 | 99.859 | 95.250 |
| | Max | 98.769 | 96.954 | 99.966 | 99.759 | 95.875 |
| Five sources each one sends two packets/seco | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 94.081 | 91.518 | 97.387 | 95.322 | 88.160 |

| | | | | | | |
|---|---|---|---|---|---|---|
| nd | Max | 94.819 | 93.076 | 98.393 | 96.313 | 88.738 |
| Five sources each one sends four | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| packets/seco | Min | 89.377 | 86.942 | 92.518 | 90.556 | 85.244 |
| nd | Max | 90.078 | 88.422 | 93.474 | 91.497 | 85.777 |
| Five sources each one sends six | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| packet/seco | Min | 84.461 | 82.160 | 87.429 | 85.576 | 79.059 |
| nd | Max | 85.124 | 83.559 | 88.333 | 86.465 | 79.544 |
| Ten sources each one sends one | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| packet/seco | Min | 94.532 | 91.927 | 98.794 | 95.894 | 90.986 |
| nd | Max | 95.892 | 93.156 | 99.205 | 96.645 | 91.854 |
| Ten sources each one sends two | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| packets/seco | Min | 95.222 | 92.627 | 97.702 | 96.478 | 88.444 |
| nd | Max | 95.968 | 94.204 | 97.926 | 97.480 | 89.014 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 91.345 | 87.256 | 96.992 | 93.787 | 78.926 |
| | Max | 92.052 | 88.741 | 98.120 | 94.202 | 79.461 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 79.403 | 75.451 | 86.280 | 81.783 | 68.255 |
| | Max | 80.018 | 76.735 | 87.255 | 82.145 | 68.718 |
| Fifteen sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 96.077 | 92.258 | 99.802 | 98.584 | 83.793 |
| | Max | 96.821 | 93.829 | 100.026 | 99.582 | 84.361 |
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 86.995 | 83.101 | 95.135 | 89.320 | 75.333 |
| | Max | 87.668 | 84.515 | 96.209 | 89.716 | 75.844 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 78.771 | 74.852 | 85.638 | 81.134 | 67.733 |
| | Max | 79.381 | 76.126 | 86.605 | 81.493 | 68.192 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 71.325 | 67.422 | 77.089 | 73.698 | 61.453 |
| | Max | 71.877 | 68.570 | 77.959 | 74.024 | 61.869 |

# APPENDIX D: 95% CONFIDENCE INTERVEL OF END-TO-END DELAY WHEN α=0.50

**Table D-1: Confidence Interval of End-to-End Delay with Pause Time=0**

| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
|---|---|---|---|---|---|---|
| | Min | 0.0672 | 0.0733 | 0.0555 | 0.0601 | 0.0817 |
| | Max | 0.0687 | 0.0744 | 0.0570 | 0.0612 | 0.0843 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.0847 | 0.0923 | 0.0699 | 0.0757 | 0.0981 |
| | Max | 0.0866 | 0.0938 | 0.0718 | 0.0771 | 0.1012 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.1090 | 0.1187 | 0.0899 | 0.0974 | 0.1252 |
| | Max | 0.1114 | 0.1206 | 0.0924 | 0.0991 | 0.1292 |
| Five sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.1402 | 0.1531 | 0.1162 | 0.1253 | 0.1603 |
| | Max | 0.1443 | 0.155 | 0.1192 | 0.1284 | 0.1652 |
| Ten sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.1401 | 0.1527 | 0.1156 | 0.1252 | 0.1668 |
| | Max | 0.1432 | 0.1551 | 0.1188 | 0.1275 | 0.1721 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.2919 | 0.3181 | 0.2408 | 0.2608 | 0.3270 |
| | Max | 0.2983 | 0.3231 | 0.2474 | 0.2656 | 0.3374 |
| Ten sources each one sends four | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| packets/second | Min | 0.3699 | 0.4084 | 0.3174 | 0.3348 | 0.4345 |
| | Max | 0.3781 | 0.4148 | 0.3262 | 0.3410 | 0.4484 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.4906 | 0.5548 | 0.4281 | 0.4449 | 0.6575 |
| | Max | 0.5015 | 0.5635 | 0.4399 | 0.4530 | 0.6785 |
| Fifteen sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.3317 | 0.3614 | 0.2736 | 0.2964 | 0.4160 |
| | Max | 0.3390 | 0.3671 | 0.2812 | 0.3018 | 0.4293 |
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.4252 | 0.4694 | 0.3648 | 0.3849 | 0.5431 |
| | Max | 0.4346 | 0.4768 | 0.3749 | 0.3919 | 0.5605 |
| Fifteen sources each one sends four | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.5452 | 0.6096 | 0.4715 | 0.4998 | 0.7147 |

| | | | | | | |
|---|---|---|---|---|---|---|
| packets/second | Max | 0.5572 | 0.6192 | 0.4849 | 0.5090 | 0.7375 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.6989 | 0.7917 | 0.6081 | 0.6491 | 0.9529 |
| | Max | 0.7144 | 0.8042 | 0.6249 | 0.6610 | 0.9833 |

**Table D-2: Confidence Interval of End-to-End Delay with Pause Time=300**

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.0183 | 0.0194 | 0.0170 | 0.0180 | 0.0248 |
| | Max | 0.0186 | 0.0198 | 0.0175 | 0.0186 | 0.0252 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.0319 | 0.0345 | 0.0281 | 0.0318 | 0.0462 |
| | Max | 0.0328 | 0.0354 | 0.0349 | 0.0326 | 0.0470 |
| Five sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.0296 | 0.0314 | 0.0276 | 0.0292 | 0.0384 |
| | Max | 0.0301 | 0.0321 | 0.0283 | 0.0301 | 0.0390 |
| Five sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.0382 | 0.041 | 0.036 | 0.0382 | 0.049 |
| | Max | 0.0394 | 0.0419 | 0.0369 | 0.0398 | 0.0508 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Ten sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.0381 | 0.0404 | 0.0355 | 0.0375 | 0.0539 |
| | Max | 0.0387 | 0.0413 | 0.0364 | 0.0387 | 0.0548 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.1586 | 0.1684 | 0.1479 | 0.1562 | 0.1926 |
| | Max | 0.1611 | 0.1722 | 0.1515 | 0.1614 | 0.1957 |
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.2397 | 0.2572 | 0.2142 | 0.2308 | 0.3007 |
| | Max | 0.2434 | 0.2630 | 0.2194 | 0.2385 | 0.3057 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.3298 | 0.3598 | 0.2793 | 0.3089 | 0.4202 |
| | Max | 0.3349 | 0.3679 | 0.2860 | 0.3192 | 0.4271 |
| Fifteen sources each one sends | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| one packet/second | Min | 0.2049 | 0.2175 | 0.1910 | 0.2017 | 0.2504 |
| | Max | 0.2081 | 0.2223 | 0.1956 | 0.2084 | 0.2545 |
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.2755 | 0.2957 | 0.2462 | 0.2653 | 0.3417 |
| | Max | 0.2798 | 0.3023 | 0.2521 | 0.2742 | 0.3474 |
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.3705 | 0.4020 | 0.3174 | 0.3490 | 0.4669 |
| | Max | 0.3763 | 0.4110 | 0.3250 | 0.3607 | 0.4746 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.4983 | 0.5465 | 0.4091 | 0.4591 | 0.6466 |
| | Max | 0.5061 | 0.5588 | 0.4189 | 0.4744 | 0.6573 |

# APPENDIX E: 95% CONFIDENCE INTERVAL OF ROUTING OVERHEAD WHEN α=0.50

**Table E-1: Confidence Interval of Routing Overhead with Pause Time=0**

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.8496 | 1.8582 | 1.7934 | 1.8015 | 1.7387 |
| | Max | 1.8639 | 1.8700 | 1.8044 | 1.8291 | 1.7644 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 2.0530 | 2.0626 | 1.9907 | 1.9997 | 1.9265 |
| | Max | 2.0689 | 2.0758 | 2.0028 | 2.0303 | 1.9549 |
| Five sources each one sends four packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 2.3037 | 2.3145 | 2.2338 | 2.2439 | 2.1593 |
| | Max | 2.3216 | 2.3293 | 2.2474 | 2.2783 | 2.1911 |
| Five sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 2.6081 | 2.6202 | 2.5289 | 2.5403 | 2.4417 |
| | Max | 2.6283 | 2.6369 | 2.5443 | 2.5792 | 2.4777 |
| Ten sources each one sends one | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| packet/second | Min | 2.4661 | 2.4776 | 2.3913 | 2.4021 | 2.1456 |
| | Max | 2.4852 | 2.4934 | 2.4059 | 2.4389 | 2.1758 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 2.0050 | 2.0143 | 1.9441 | 1.9529 | 1.7047 |
| | Max | 2.0205 | 2.0272 | 1.9560 | 1.9828 | 1.7298 |
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.8309 | 1.8395 | 1.7745 | 1.7826 | 1.7463 |
| | Max | 1.8452 | 1.8514 | 1.7855 | 1.8103 | 1.7725 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.1051 | 1.1186 | 1.0190 | 1.0326 | 0.9693 |
| | Max | 1.1269 | 1.1366 | 1.0356 | 1.0745 | 1.0088 |
| Fifteen sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 2.1105 | 2.1203 | 2.0464 | 2.0557 | 2.0055 |
| | Max | 2.1269 | 2.1339 | 2.0589 | 2.0872 | 2.0351 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 2.0805 | 2.0903 | 2.0164 | 2.0257 | 1.9755 |
| | Max | 2.0969 | 2.1039 | 2.0289 | 2.0572 | 2.0051 |
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.1074 | 1.1219 | 1.0205 | 1.0342 | 0.9751 |
| | Max | 1.1291 | 1.1400 | 1.0372 | 1.0763 | 1.0149 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.4765 | 1.4979 | 1.3589 | 1.3790 | 1.3142 |
| | Max | 1.5055 | 1.5221 | 1.3811 | 1.4350 | 1.3678 |

**Table E-2: Confidence Interval of Routing overhead with Pause Time=300**

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.3476 | 1.3822 | 1.3033 | 1.3102 | 1.2522 |
| | Max | 1.3987 | 1.4300 | 1.3337 | 1.3494 | 1.2766 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.4958 | 1.5342 | 1.4467 | 1.4543 | 1.3886 |
| | Max | 1.5526 | 1.5872 | 1.4804 | 1.4978 | 1.4156 |
| Five sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.6785 | 1.7216 | 1.6233 | 1.6319 | 1.5745 |
| | Max | 1.7422 | 1.7810 | 1.6612 | 1.6807 | 1.6051 |
| Five sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.9002 | 1.9490 | 1.8378 | 1.8475 | 1.7869 |
| | Max | 1.9723 | 2.0163 | 1.8807 | 1.9027 | 1.8216 |
| Ten sources each one sends one | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| packet/second | Min | 1.7968 | 1.8429 | 1.7378 | 1.7469 | 1.6808 |
| | Max | 1.8650 | 1.9066 | 1.7783 | 1.7992 | 1.7135 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.4608 | 1.4983 | 1.4128 | 1.4203 | 1.4007 |
| | Max | 1.5162 | 1.5501 | 1.4458 | 1.4628 | 1.4279 |
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.3796 | 1.4143 | 1.3258 | 1.3420 | 1.3028 |
| | Max | 1.4309 | 1.4623 | 1.3564 | 1.3814 | 1.3277 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.8220 | 0.8748 | 0.7083 | 0.7655 | 0.6997 |
| | Max | 0.8996 | 0.9473 | 0.7554 | 0.8250 | 0.7359 |
| Fifteen sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.5377 | 1.5772 | 1.4766 | 1.4950 | 1.4744 |
| | Max | 1.5960 | 1.6317 | 1.5114 | 1.5397 | 1.5031 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.5677 | 1.6072 | 1.5066 | 1.5250 | 1.5044 |
| | Max | 1.6260 | 1.6617 | 1.5414 | 1.5697 | 1.5331 |
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 0.8236 | 0.8774 | 0.7094 | 0.7667 | 0.7018 |
| | Max | 0.9014 | 0.9502 | 0.7565 | 0.8263 | 0.7381 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 1.0981 | 1.1714 | 0.9446 | 1.0222 | 0.9114 |
| | Max | 1.2019 | 1.2686 | 1.0074 | 1.1018 | 0.9586 |

# APPENDIX F: 95% CONFIDENCE INTERVAL OF ENERGY

## CONSUMPTION PERCENTAGE WHEN α=0.50

**Table F-1: Confidence Interval of Energy Consumption Percentage with Pause Time=0**

| | | | | | | |
|---|---|---|---|---|---|---|
| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 21.260 | 22.016 | 19.120 | 19.736 | 27.272 |
| | Max | 21.895 | 22.621 | 19.835 | 20.391 | 27.879 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 23.811 | 24.658 | 21.414 | 22.105 | 30.408 |
| | Max | 24.522 | 25.335 | 22.215 | 22.838 | 31.085 |
| Five sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 28.335 | 29.343 | 25.483 | 26.305 | 34.027 |
| | Max | 29.181 | 30.149 | 26.436 | 27.177 | 34.784 |
| Five sources each one sends six | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| packets/second | Min | 34.568 | 34.918 | 30.324 | 31.303 | 41.172 |
| | Max | 35.601 | 35.878 | 31.458 | 32.340 | 42.089 |
| Ten sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 27.256 | 28.226 | 24.512 | 25.303 | 34.090 |
| | Max | 28.070 | 29.001 | 25.429 | 26.142 | 34.848 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 38.388 | 38.665 | 36.586 | 36.671 | 49.085 |
| | Max | 39.536 | 39.728 | 37.954 | 37.887 | 50.178 |
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 59.099 | 59.803 | 56.494 | 56.968 | 68.142 |
| | Max | 60.866 | 61.447 | 58.607 | 58.857 | 69.658 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 61.584 | 62.983 | 58.240 | 59.357 | 70.547 |
| | Max | 63.424 | 64.714 | 60.418 | 61.324 | 72.116 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Fifteen sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 46.216 | 49.451 | 43.905 | 46.014 | 59.216 |
| | Max | 47.687 | 50.813 | 45.659 | 47.573 | 60.455 |
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 55.927 | 58.888 | 53.483 | 55.006 | 68.241 |
| | Max | 57.599 | 60.443 | 55.483 | 56.793 | 69.649 |
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 63.554 | 67.598 | 60.984 | 62.930 | 74.221 |
| | Max | 65.453 | 69.373 | 63.265 | 64.979 | 75.815 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 72.220 | 73.742 | 69.537 | 70.906 | 82.351 |
| | Max | 74.378 | 75.768 | 72.138 | 73.256 | 84.202 |

**Table F-2: Confidence Interval of Energy Consumption Percentage with Pause Time=300**

| Five sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
|---|---|---|---|---|---|---|
| | Min | 15.135 | 16.246 | 13.820 | 14.315 | 19.951 |
| | Max | 15.508 | 16.652 | 14.103 | 14.607 | 20.406 |
| Five sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 16.952 | 18.195 | 15.478 | 16.032 | 22.644 |
| | Max | 17.369 | 18.650 | 15.795 | 16.360 | 23.161 |
| Five sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 20.172 | 21.652 | 18.419 | 19.079 | 25.543 |
| | Max | 20.669 | 22.193 | 18.796 | 19.469 | 26.126 |
| Five sources each one sends six packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 24.610 | 25.766 | 21.919 | 22.703 | 31.341 |
| | Max | 25.216 | 26.410 | 22.367 | 23.168 | 32.056 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Ten sources each one sends one packet/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 19.404 | 20.828 | 17.718 | 18.352 | 27.330 |
| | Max | 19.882 | 21.348 | 18.080 | 18.727 | 27.954 |
| Ten sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 27.330 | 28.531 | 26.445 | 26.597 | 39.374 |
| | Max | 28.002 | 29.244 | 26.985 | 27.141 | 40.273 |
| Ten sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 54.369 | 56.061 | 50.244 | 53.210 | 60.807 |
| | Max | 55.707 | 57.462 | 51.272 | 54.298 | 62.055 |
| Ten sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 55.504 | 56.515 | 54.324 | 54.637 | 68.417 |
| | Max | 56.869 | 57.928 | 55.435 | 55.755 | 69.684 |
| Fifteen sources each one sends | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |

| | | | | | | |
|---|---|---|---|---|---|---|
| one packet/second | Min | 44.325 | 46.405 | 42.791 | 43.956 | 53.507 |
| | Max | 45.489 | 47.640 | 43.728 | 44.898 | 54.956 |
| Fifteen sources each one sends two packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 54.023 | 55.699 | 51.073 | 52.877 | 64.099 |
| | Max | 55.352 | 57.091 | 52.149 | 53.958 | 65.517 |
| Fifteen sources each one sends four packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 61.671 | 62.795 | 58.360 | 60.708 | 73.183 |
| | Max | 63.188 | 64.364 | 59.594 | 61.950 | 74.576 |
| Fifteen sources each one sends six packets/second | Confidence Interval | VCAR_Q_ACC | VCAR_Q_MAX | VCAR_R_ACC | VCAR_R_MAX | AODV |
| | Min | 70.400 | 70.794 | 67.033 | 69.699 | 82.585 |
| | Max | 72.132 | 72.564 | 68.009 | 71.125 | 83.961 |