Volume 4, No. 8, May-June 2013
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Improving Efficiency of MANET by Reducing Queuing Delay Using Hybrid Algorithm

Madhup Shrivastava*[1], Monika Sahu[2], Sonika Matele[3] and Dr. M. A. Rizvi[4]
NITTTR Bhopal(M.P.)
08madhup@gmail.com*[1], monikasahu7@gmail.com[2], sonikamatele@gmail.com[3], marizvi@nitttrbpl.ac.in[4]

Abstract: A mobile ad hoc network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless and it getting popular means of data transfer these days. MANETs have become the subject of intensive research over the past decade. MANET is an autonomous system of mobile nodes and is free to move independently in any direction at varying speeds, and will therefore change its links to other devices frequently. Each node must forward traffic unrelated to its own use, and therefore be a router. Each flow from the source to the destination traverses multiple hops of wireless links. In the mobile ad hoc networks, due to the multi-hop of the data, the limited bandwidth and the dynamic changes of the network topology, the network performance is hindered. It is the need that quality of service has to be improved.

Now days these types networks are gaining popularity in the form of wi-fi zones etc. MANETs are a kind of wireless ad hoc networks that usually has a routable networking environment on top of a Link Layer ad hoc network. The need has been felt to improve the services of MANET. In this paper an attempt is made to improve the quality of service QoS of MANET by reducing queuing delay in turn increasing throughput by proposing a new hybrid algorithm.

Keywords: MANET, FCFS (First Come First Serve), SJF (Shortest Job First), Queuing delay, end-to-end delay, throughput, hybrid algorithm

I. INTRODUCTION

A MANET, sometimes called a mobile mesh network, is a self-configuring network of mobile devices connected by wireless links and centralized administration. Communication in MANET is done via multi-hop paths. Wireless communication technology is emerging day by day; with such growth sooner or later it would not be practical or simply physically possible to have a fixed architecture for this kind of network. Ad hoc wireless network must be capable to self-organize and self-configure due to the fact that the mobile structure is changing all the time [1].

A mobile ad hoc network (MANET) is a wireless network that uses multi-hop peer to-peer routing instead of static network infrastructure to provide network connectivity. MANETs have applications in rapidly deployed and dynamic military and civilian systems. The network topology in a MANET usually changes with time.

Mobile Ad hoc Networks (MANET) consist of wireless nodes that form a communications network among themselves without a fixed infrastructure [2].

A. Characteristics of MANET:

a. Communication via wireless means i.e. Nodes can perform the roles of both hosts and routers.
b. It supports dynamic topologies i.e. nodes are free to move arbitrarily
c. The most important system design criteria in MANET for optimization is energy conservation
d. Here communication environment is different from fixed network
e. Supports high density with the large number of user mobility
f. Supports different and differing radio propagation conditions throughout the network
g. Supports intermittent nodal connectivity
h. Wireless links are particularly vulnerable to the attacks [3]

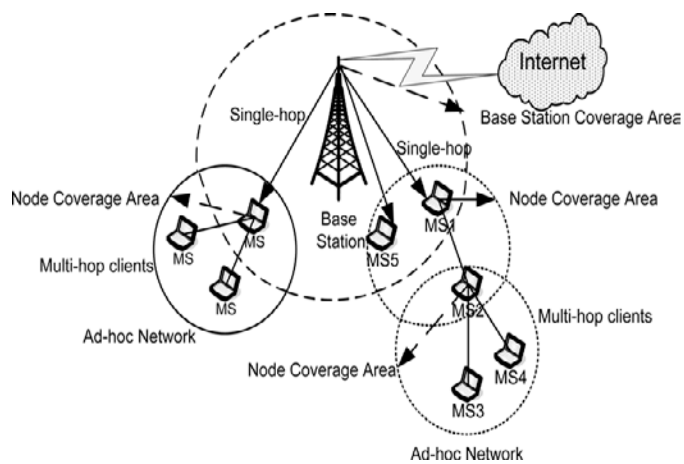

Figure 1: Ad-hoc Network

B. Issues in MANETs: MANET is very powerful network though it has lots of challenges such as:

a. MANET contains diverse resources
b. The line of defense is very ambiguous
c. Nodes operate in shared wireless medium
d. Network topology changes unpredictably and very dynamically
e. Radio link reliability is an issue
f. Frequent connection breaks down.

Moreover, density of nodes, number of nodes and mobility of these hosts may vary in different applications. There is no stationary infrastructure. Each node in MANET acts a router that forwards data packets to other nodes. Therefore, selection of effective, suitable, adaptive and robust routing protocol is of utmost importance. [4]

If there are a number of mobile hosts wishing to communicate, then the routing protocols come into picture, in this case some critical decisions have to be made such as which is the optimal route from the source to the destination which is very important because, the mobile nodes operate on battery power. Thus it becomes necessary to transfer the data with the minimal delay to loss less power. In addition to this, Quality of Service support is also needed so that the least packet drop can be obtained. The other factors which need to be considered while choosing a protocol for MANETs are as follows:

- ***a)*** ***Multicasting:*** The ability to send packets to multiple nodes at once. This is important as it takes less time to transfer data to multiple nodes.
- ***b)*** ***Loop Free:*** A path taken by a packet never transits the same intermediate node twice before it arrives at the destination. To improve the overall performance in the routing protocol to guarantee that the routes supplied are loop-free. This avoids any loss of bandwidth or CPU consumption.
- ***c)*** ***Multiple routes:*** If one route gets broken due to some disaster, then the data could be sent through some other route. Thus the protocol should allow creating multiple routes.
- ***d)*** ***Distributed Operation:*** The protocol should be distributed. It should not be dependent on a centralized node.
- ***e)*** ***Reactive:*** It means that the routes are discovered between a source and destination only when the need arises to send data.
- ***f)*** ***Unidirectional Link Support:*** The radio environment can cause the formation of unidirectional links. Utilization of these links and not only the bi-directional links improves the routing protocol performance.
- ***g)*** ***Power Conservation:*** The nodes in an ad-hoc network can be laptops and thin clients, such as PDAs that are very limited in battery power and therefore use some sort of standby mode to save power. It is therefore important that the routing protocol has support for these sleep-modes. [5]

II. QOS PARAMETERS IN MANET

In the field of computer networking and other packet switched telecommunication networks, the traffic engineering term quality of service (QoS) refers to resource reservation control mechanisms rather than the achieved service quality. QoS is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow.

Many things can happen to packets as they travel from origin to destination, resulting in the following problems as seen from the point of view of the sender and receiver in MANET:

A. Throughput:

Due to varying load from other users sharing the same network resources, the bit rate (the maximum throughput) that can be provided to a certain data stream may be too low for real time multimedia services if all data streams get the same scheduling priority.

B. Packets Drop:

The routers might fail to deliver (drop) some packets if their data is corrupted or they arrive when their buffers are already full.

C. Errors:

Sometimes packets are corrupted due to bit errors caused by noise and interference, especially in wireless communications.

D. Latency:

It might take a long time for each packet to reach its destination, because it gets held up in long queues, or takes a less direct route to avoid congestion. This is different from throughput, as the delay can build up over time, even if the throughput is almost normal.

E. Jitter:

A packet's delay varies with its position in the queues of the routers along the path between source and destination and this position can vary unpredictably. This variation in delay is known as jitter and can seriously affect the quality of streaming audio and/or video.

F. Out-of-order delivery:

When a collection of related packets is routed through a network, different packets may take different routes, each resulting in a different delay. The result is that the packets arrive in a different order than they were sent. This problem requires special additional protocols responsible for rearranging out-of-order packets to an isochronous state once they reach their destination [6].

G. Packet Delivery Ratio:

It is the number of packets received by the destination upon the number of packets generated by the source. It specifies the Packet loss rate, which limits the maximum throughput of the network.

H. End-to-end delay:

It can be defined as the time taken for a packet to be transmitted in a network from source to destination.

III. ANALYSIS OF VARIOUS DELAYS IN END TO END DELAY

Mathematically End to End delay is shown below:

$d_{end-end}= N[\, d_{trans}+d_{prop}+d_{proc}]$

where

$d_{end-end}$= end-to-end delay

d_{trans}= transmission delay

d_{prop}= propagation delay

d_{proc}= processing delay

N= number of links (Number of routers + 1)

A. Transmission delay:

In a network based on packet switching, transmission delay (or store-and-forward delay) is the amount of time required to push all of the packet's bits into the wire. In other words, this is the delay caused by the data-rate of the link.

Transmission delay is a function of the packet's length and has nothing to do with the distance between the two nodes. This delay is proportional to the packet's length in bits,
It is given by the following formula:

$$D_T = N/R$$

Where,

D_T is the transmission delay

N is the number of bits, and

R is the rate of transmission (say in bits per second)

B. Propagation delay:

In computer networks, **propagation delay** is the amount of time it takes for the head of the signal to travel from the sender to the receiver. It can be computed as the ratio between the link length and the propagation speed over the specific medium.

Propagation delay is equal to d / s where d is the distance and s is the wave propagation speed. In wireless communication, s=c, i.e. the speed of light. In copper wire, the speed (s) generally ranges from .59c to .77c. This delay is the major obstacle in the development of high-speed computers and is called the interconnect bottleneck in IC systems.

C. Processing delay:

In a network, based on packet switching, processing delay is the time which takes by routers to process the packet header. Processing delay is a key component in network delay. During processing of a packet, routers may check for bit-level errors in the packet that occurred during transmission as well as determining where the packet's next destination is. Processing delays in high-speed routers are typically on the order of microseconds or less. After this nodal processing, the router directs the packet to the queue where further delay can happen (queuing delay). [7]

D. Packet transfer delay:

Packet transfer delay is a concept in packet switching technology. The sum of store-and-forward delay that a packet experiences in each router gives the transfer or queuing delay of that packet across the network. Packet transfer delay is influenced by the level of network congestion and the number of routers along the way of transmission.

There are four sources of packet transfer delay:

a. Nodal processing:
   a) Check bit errors
   b) Determine output link
b. Queuing:
   a) Time waiting at output link for transmission
   b) Depends on congestion level of router
c. Transmission delay:
   a) R=Link bandwidth (bit/s)
   b) L=Packet length (bits)
   c) Time to send bits into link = L/R
d. Propagation delay:
   a) d = Length of physical link
   b) s = Propagation speed in medium
   c) Propagation delay = d/s

E. Queuing in Packet Transfer Delay:

Network delays are usually small. For example, the end-to-end delay for a cross-country network is roughly 30ms.

However, packets can get lost in a network. Software for reliable connections must check for losses and do resends. If a resend is needed, the overall delay is at least doubled: the another round-trip time is added for a resend request and response. For higher speed reliable data transfer protocols the impact can be even greater.

$$d_{end\text{-}to\text{-}end} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

The queuing delay dqueue is the time that a packet spends in a queue at a node while waiting for other packets to be transmitted. If the node is a high-speed router then there is one queue for each outgoing link, so a packet waits only for other packets that are going across the same link.

The queuing delay is related to the transmission delay dtrans by the following approximate equation.

$$d_{queue} = d_{trans} * l_{queue}$$

Here, lqueue is the average length of the queue. The average queue length depends on the load factor, which is the ratio of the attempted link transmission rate to the link maximum transmission rate. The average queue length is typically less than 1 for a load factor less than 1/2. When the load factor exceeds 1, the queue length grows without bound. [8]

In telecommunication and computer engineering, the queuing delay (or queuing delay) is the time a job waits in a queue until it can be executed. It is a key component of network delay. In a switched network, the time between the completion of signaling by the call originator and the arrival of a ringing signal at the call receiver. Queues may be caused by delays at the originating switch, intermediate switches, or the call receiver servicing switch.

This term is most often used in reference to routers. When packets arrive at a router, they have to be processed and transmitted. A router can only process one packet at a time. If packets arrive faster than the router can process them (such as in a burst transmission) the router puts them into the queue (also called the buffer) until it can get around to transmitting them. Delay can also vary from packet to packet so averages and statistics are usually generated when measuring and evaluating queuing delay.

As a queue begins to fill up due to traffic arriving faster than it can be processed, the amount of delay a packet experiences going through the queue increases. The speed at which the contents of a queue can be processed is a function of the transmission rate of the facility. This leads to the classic delay curve. The average delay any given packet is likely to experience is given by the formula $1(\mu\text{-}\lambda)$ where μ is the number of packets per second the facility can sustain and λ is the average rate at which packets are arriving to be serviced. This formula can be used when no packets are dropped from the queue.

The maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is. The router queue of packets waiting to be sent also introduces a potential cause of packet loss. When the buffer fills the router must drop packets to remain functional, resulting in data loss. Since the router has a finite amount of buffer memory to hold the queue, a router which receives packets at too high a rate may experience a full queue. In this case, the router has no other option than to simply discard excess packets. If required, these may later be retransmitted by a transport protocol.

When the transmission protocol uses the dropped-packets symptom of filled buffers to regulate its transmit rate, as the Internet's TCP does, bandwidth is fairly shared at near theoretical capacity with minimal network congestion delays. Absent this feedback mechanism the delays become both unpredictable and rise sharply, a symptom also seen as freeways approach capacity; metered onramps are the most effective solution there, just as TCP's self-regulation is the most effective solution when the traffic is packets instead of cars). This result is both hard to model mathematically and quite counterintuitive to people who lack experience with mathematics or real networks. Failing to drop packets, choosing instead to buffer an ever-increasing number of them, produces bufferbloat (is a phenomenon in a packet-switched computer network where by excess buffering of packets inside the network causes high latency and jitter, as well as reducing the overall network throughput).

There are three components describing the behavior of a queue:

a. The customers arriving for service, which is usually described by a Poisson process (random arrivals), but sometimes by non-Poisson processes or even deterministic arrivals rates
b. The time required to service each customer, which is usually described by a probability distribution, e.g. exponential or gamma (Erlang) distributed service times, possibly deterministic though.
c. The number of service providers, a positive integer value.

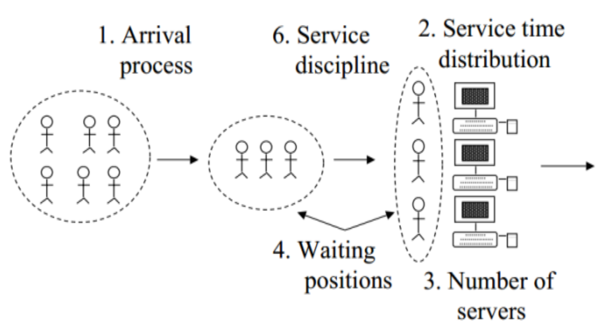

Figure 2: General processing in a queue

The basic queuing model is shown in figure 2. It can be used to model, e.g., machines or operators processing orders or communication equipment processing information.

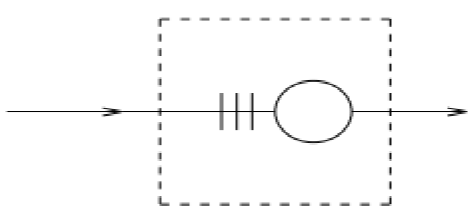

Figure 3: Basic queuing model

Among others, a queuing model is characterized by:

a) The arrival process of customers:

Usually we assume that the interarrival times are independent and have a common distribution. In many practical situations customers arrive according to a Poisson stream (i.e. exponential interarrival times). Customers may arrive one by one, or in batches.

b) The service times:

Usually we assume that the service times are independent and identically distributed, and that they are independent of the interarrival times.

c) The service discipline:

Customers can be served one by one or in batches. We have many possibilities for the order in which they enter service. We mention:

(a). first come first served, i.e. in order of arrival;
(b). random order;
(c). last come first served (e.g. in a computer stack or a shunt buffer in a production line);
(d). priorities (e.g. rush orders first, shortest processing time first);
(e). Processor sharing (in computers that equally divide their processing power over all jobs in the system).

d) The service capacity:

There may be a single server or a group of servers helping the customers.

e) The waiting room:

There can be limitations with respect to the number of customers in the system. For example, in a data communication network, only finitely many cells can be buffered in a switch. The determination of good buffer sizes is an important issue in the design of these networks.

IV. ANALYSIS OF USED "QUEUING ALGORITHMS"

The queuing in packet transfer delay generally and mostly uses the FCFS for their processing of jobs and some delays are found when the network is congested. Sometimes it causes the deadlock kind of condition in the network, deadlock cannot occur in the network because of the time to live field of the packet. But it causes the network delay twice once for the waiting and if the packet is lost then second for the retransmission. The queuing follows the concept of FCFS, Round Robin etc.

A. FCFS (First come, first served):

First come, first served (FCFS) is a scheduling algorithm and a network routing management mechanism that automatically executes queued requests and processes by the order of their arrival. With first come, first served, what comes first is handled first; the next request in line will be executed once the one before it is complete.[9]

A FIFO acts like any normal queue whether, it is a line in a cinema, a checkout line in a store, or a queue for a ticket booth. The first person or process to arrive (First In) is the first one to be dealt with (First Out). If one person goes through the line and then decides they forgot something then they have to go back through.

a. Implementation:

To implement this, you can create a queue, an abstract data type (ADT) that can be constructed from a linked list data structure. The system can dequeue the next process from the front of the queue, run the process until completion (or enqueue the process at the end of the line in more complex schemes), then enqueue the process at the end of the line, allowing the next process to use the CPU.

b. Advantages:

a) Simple
b) Easy to understand
c) First come, first served

c. Disadvantages:

a) This scheduling method is nonpreemptive, that is, the process will run until it finishes.
b) Because of this nonpreemptive scheduling, short processes which are at the back of the queue have to wait for the long process at the front to finish

B. SJF (Shortest Job First):

Other name of this algorithm is Shortest-Process-Next (SPN). Shortest-Job-First (SJF) is a non-preemptive discipline in which waiting job (or process) with the smallest estimated run-time-to-completion is run next. In other words, when CPU is available, it is assigned to the process that has smallest next CPU burst. The SJF scheduling is especially appropriate for batch jobs for which the run times are known in advance. Since the SJF scheduling algorithm gives the minimum average time for a given set of processes, it is probably optimal. The SJF algorithm favors short jobs (or processors) at the expense of longer ones.[9]**.**

In the production environment where the same jobs run regularly, it may be possible to provide reasonable estimate of run time, based on the past performance of the process. But in the development environment users rarely know how their program will execute. Like FCFS, SJF is non preemptive therefore, it is not useful in timesharing environment in which reasonable response time must be guaranteed. [10]

a. Advantage:

a) SJF is optimal – gives minimum average waiting time for a given set of processes
b) It gives superior turnaround time performance to shortest process next because a short job is given immediate preference to a running longer job.
(a). Throughput is high.
(b). Disadvantage
(c). Elapsed time (i.e., execution-completed-time) must be recorded, it results an additional overhead on the processor.
(d). Starvation may be possible for the longer processes.[11]

C. Priority Scheduling:

The SJF is a special case of general priority scheduling algorithm. A Priority (an integer) is associated with each process. The CPU is allocated to the process with the highest priority. Generally smallest integer is considered as the highest priority. Equal priority processes are scheduled in First Come First serve order. It can be preemptive or Non-preemptive.

a. Non-preemptive Priority Scheduling:

In this type of scheduling the CPU is allocated to the process with the highest priority after completing the present running process.

Advantage

(a). Good response for the highest priority processes.

Disadvantage

(a). Starvation may be possible for the lowest priority processes.

b. Preemptive Priority Scheduling:

In this type of scheduling the CPU is allocated to the process with the highest priority immediately upon the arrival of the highest priority process. If the equal priority process is in running state, after the completion of the present running process CPU is allocated to this even though one more equal priority process is to arrive.

Advantage

(a). Very good response for the highest priority process over non-preemptive version of it.

Disadvantage

(a). Starvation may be possible for the lowest priority processes.

V. PROPOSED ALGORITHM

To remove the deficiencies of the algorithms exist for the queuing in the MANET and minimize the delay we tried to propose an algorithm, which is:

The FCFS algorithm is simple by it process the job first which has come first, but it is nonpreemptive by its nature too, so the process which comes last will suffer from the problem of waiting if the process which is processing is heavy in processing. This deficiency was removed by the SJF(Shortest Job First) algorithm .SJF is optimal, it gives minimum average waiting time for a given set of processes but need to have a good heuristic to guess the next CPU execution time because before the processing it requires the execution time to schedule the sequence of the processes. It is also nonpreemptive by nature. These deficiencies are removed by preemptive algorithms of scheduling. Priority algorithm is one which removes the deficiency of FCFS and SJF. Priority algorithm has good response for the highest priority processes but starvation may be possible for the lowest priority processes.

There are several different criteria to consider when trying to select the "best" scheduling algorithm for a particular situation and environment, including:

a. CPU utilization - Ideally the CPU would be busy 100% of the time, so as to waste 0 CPU cycles. On a real system

CPU usage should range from 40% (lightly loaded) to 90% (heavily loaded)

b. ***Throughput*** **-** Number of processes completed per unit time. May range from 10 / second to 1 / hour depending on the specific processes.

c. ***Turnaround time*** **-** Time required for a particular process to complete, from submission time to completion. (Wall clock time)

d. ***Waiting time*** **-** How much time processes spend in the ready queue waiting their turn to get on the CPU.

- ***a)*** *(**Load average*** **-** The average number of processes sitting in the ready queue waiting their turn to get into the CPU. Reported in 1-minute, 5-minute, and 15-minute averages by "uptime" and "who".)

e. ***Response time*** **-** The time taken in an interactive program from the issuance of a command to the commence of a response to that command.

In general one wants to optimize the average value of criteria. However sometimes one wants to do something different, such as to minimize the maximum response time. Sometimes it is most desirable to minimize the ***variance*** of criteria than the actual value. I.e. users are more accepting of a consistent predictable system than an inconsistent one, even if it is a little bit slower. [12]

A. Algorithm Developed:

a. Sense the channel using CSMA/CA. By this we can sense the traffic over the channel.

Now 4 conditions will arises:

a) The channel is free means we can send the data over the channel easily.

b) The channel is quite busy means we can send the data but the reliability of data receiving will be less.

c) The channel is too busy means the reliability is very less.

d) The sensing of the channel is failed.

b. Apply the hybrid algorithm which is the combination of FCFS, SJF and Priority algorithm of scheduling.

```
If (Channel == free)
{
//Apply FCFS for scheduling the processes into the channel.
Get the burst and arrival time of all processes and push into buffer
In increase order with respect to arrival time
Now pop the process come first and process it.
}
else if(Channel == busy)
{
//Apply the SJF algorithm of scheduling for the processes.
Get the burst and arrival time of all processes and push into buffer
In increase order with respect to burst time
Now pop the process come first and process it.
}
else if (Channel == very busy)
{
//Apply the Priority scheduling with preemption to schedule the processes.
Get the burst and arrival time and set priority of all processes and push into buffer
In increase order with respect to priority and arrival time
Now pop the process come first and process it.
}
else
{
Again sense the channel
}
```

B. Flow Chart:

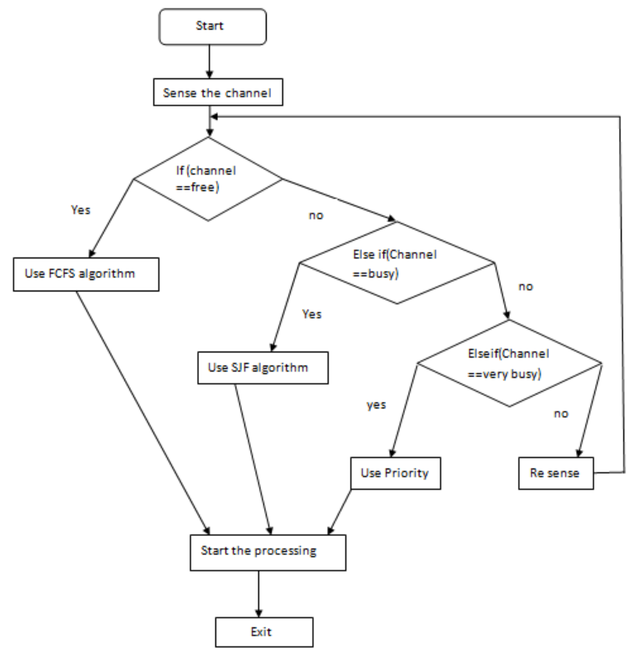

Figure 5: Flow Chart of Proposed Algorithm

VI. CONCLUSION

As mobile devices and wireless communications have burgeoned at an aberrant rate, MANETs have become the subject of intensive research over the previous decennium. In this paper, the depiction is given about the importance of QoS of MANET. Issues related with Various QoS parameter of Mobile Ad-Hoc networks, like end to end delay, packet delivery ratio, packet drop, throughput etc. were analyzed along with the algorithms used.

The ends-to- end delay of packet over the network is espoused and try to increase the efficiency of the network. In end-to-end delay the focus is on Queuing delay of packets on the nodes and tried to reduce is delay using newly developed hybrid algorithm. The algorithm is using the CSMA/CA for sensing the traffic level of channel and the next working of algorithm depends upon that traffic level.

This new developed hybrid algorithm will enhance the efficiency in terms of reduced end-to-end delay of packet and in turn throughput will be increased.

VII. REFERENCES

[1]. "Comparing AODV and OLSR Routing Protocols" Aleksandr Huhtonen Helsinki University of Technology Telecommunication Software and Multimedia Laboratory

[2]. Jailani Kadir, Osman Ghazali, Mohamed Firdhous, Suhaidi Hassan"Node Selection Based on Energy Consumption in MANET" InterNetWorks Research Group, School of Computing, University Utara Malaysia, Malaysia

[3]. Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt and Piet Demeester "An Overview of Mobile Ad Hoc Networks: Applications and Challenges"

[4]. Jatin D. Parmar Ashish D. Patel Rutvij H. Jhaveri Bhavin I. Shah"MANET Routing Protocols and Wormhole Attack against AODV" (IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010

[5]. R.balakrishna, s.jeyabalan, dr.u.rajeswar rao, dr.t.k.basak4, dr. V.cyrilraj "Performance issues on aodv and dsdv for Manets" (Journal of Theoretical and Applied Information Technology 2005 - 2010)

[6]. Vishal Garg, Chandan Kapoor"A Survey of QoS parameters through reactive routing in MANETs" (IJCEM International Journal of Computational Engineering & Management, Vol. 13, July 2011 ISSN (Online): 2230-7893)

[7]. en.wikipedia.org/wiki/Queuing_delay

[8]. http://www.d.umn.edu/~gshute/net/new/delays-losses.x

[9]. http://www.techopedia.com/definition/23455/first-come-first-served-fcfs

[10]. http://www.personal.kent.edu/~rmuhamma/OpSystems/Myos / sjfSchedule.htm

[11]. "Queueing Theory" Ivo Adan and Jacques Resing Department of Mathematics and Computing Science Eindhoven University of Technology, The Netherlands February 28, 2002

[12]. http://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/ 5_ CPU _Scheduling.html

[13]. http://www.go4expert.com/forums/showthread.php?t=22307

[14]. http://en.wikipedia.org/wiki/Shortest_job_next