Network and Complex Systems www.iiste.org
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.5, No.9, 2015

Congestion Control in Mobile Ad Hoc Network using modified acknowledgement with secure channel

Khushboo Sharma
Department of Computer Science and Technology
Radharaman Institute of Technology and Science
Madhya Pradesh,Bhopal,India
Anurag Jain
Department of Computer Science and Technology
Radharaman Institute of Technology and Science
Madhya Pradesh,Bhopal,India

ABSTRACT

The mobile ad hoc network is self-configuring and dynamic in nature. Due to its dynamic topology node can join or leave any time and each node behaves as router or host which can deliver the packets from source to destination. Due to the heavy traffic load over network congestion occur. To avoid the congestion on network various congestion control mechanism has been developed but in this we use modified-ACK based scheme for node authentication in AODV protocol. The simulation of our proposed work is done on network simulator NS-2.34 and comparative analysis of our proposed methodology is done using performance metrics such as packet delivery ratio, throughput, end-end delay average jitter and routing load.

Keywords

MANET, Congestion Control, AODV, ACK, PDR, Network Simulator

1. INTRODUCTION

Ad hoc network is a cluster of mobile node. During the past few years we have all witnessed progressively increasing enlargement in the deployment of wireless and mobile communication networks [1]. Mobile ad hoc networks encompasses of nodes that are capable to communicate through the use of wireless mediums and form self-motivated topologies. The basic distinctive of these networks is the inclusive lack of any kind of infrastructure, and consequently the absence of dedicated nodes that make available network management operations as do the traditional routers in fixed networks. With a specific end goal to support availability in a portable impromptu system every single taking part hub need to accomplish steering of system movement .In order to sustain connectivity in a mobile ad hoc network all participating nodes have to achieve routing of network traffic. The cooperation of nodes cannot be enforced by a centralized supervision authority since one does not exist. Consequently, a network-layer protocol designed for such self-organized networks must impose connectivity and security requirements in order to assure the undisrupted operation of the higher layer protocols. Unfortunately all of the extensively used ad hoc routing protocols have no security considerations and belief all the participants to properly forward routing and data traffic. In portable impromptu systems, blockage happens with restricted assets. The customary TCP clog control component is not able to control the exceptional properties of a mutual remote channel. TCP blockage control works to a great degree well on the Internet. However portable specially appointed systems display some remarkable properties that significantly influence the outline of suitable conventions and convention stacks when all is said in done and of blockage control component specifically. As it turned out, the all that much contrasting environment in a portable impromptu system is profoundly hazardous for standard TCP [2]. TCP convention might encounter execution corruptions over remote systems, because of non-blockage related bundle misfortune and differing round excursion times unmodified standard TCP performs insufficiently in a remote situation because of its powerlessness to recognize parcel misfortunes brought on by system clog from those ascribed to transmission blunders. Parcel misfortune or reaction of out of-request bundles shows disappointments. To uproot such disappointments, TCP actualizes stream control and blockage control calculations in view of the sliding window. The execution of TCP is by and large lower in remote systems than in altered. In this paper ACK-Based plan for hub verification in AODV convention in portable specially appointed system. ACK-Based plan likewise give office to the recognition of wormhole assault and hub mischief in impromptu system. ACK based plan beat the constraint of uninvolved criticism strategy when force control transmission is utilized.The scenario of mobile ad hoc network is shown in fig. 1.

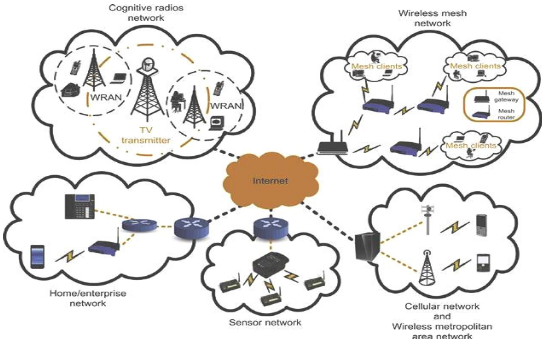


Fig. 1 Architecture of Mobile ad hoc Network

2. *CONGENTIOSN & CHALLENGES OF TCP PROTOCOL IN MANET*

2.1 Congestion in TCP Protocol

Congestion can be termed as obstruction where successful communication do not takes place. It is a situation in network communication where too many packets are present in the subnet [8]. The main reason for congestion is the presence of limited resources or when the offered load is greater than the available capacity of the channel. Congestion leads to high packet loss and bandwidth degradation and waste energy and time in its recovery [8]. Congestion can create the following difficulties:

- Long Delay: The congestion control mechanism takes much time for detecting congestion. When the congestion is more severe, it is preferred to select an alternate new path.
- High overhead: It takes into account in discovery of new routes for processing and communication. If multipath routing is used, it needs additional effort for maintaining the multipath regardless of the existence of alternate route.
- Many packet losses: Congestion leads to packet losses. To minimize the traffic load, a congestion control solution is applied either by reducing the sender rate at the sender's side or by dropping the packets at the intermediate nodes or by applying both the methods.

2.1 Challenges in TCP Protocol

TCP and MANETs both are based on the transport layer protocol for communication but MANET follows multi hop communication for data routing [9][10]. It is the atmosphere of a traditional wired network between the hosts which provides a reliable end-to-end data delivery. TCP durability is achieved by retransmitting lost packets. Therefore, the estimated round-trip delay of each TCP sender and the average deviation derived from it maintains a running average. The sender receives a receipt if any within a specified timeout interval is retransmitted packets will be duplicated or received. Dependability on the tradition wired network, no packet loss due to congestion is an implicit assumption that the TCP is made. To reduce congestion TCP congestion control [11] [12] is used when packet loss is detected. Wireless ad hoc networks are some of the features [13] of TCP performance that will significantly deteriorate if they do not do anything. Basically, these features, such as channel burst errors, mobility and communication asymmetry.

1. Channel Error

In Wireless channels, due to multipath fading and shadowing, high bit error rate of transmission occurs and it can corrupt packet transmission and may lead to loss of TCP data segments or ACK packets. If it does not receive ACK within timeout retransmissions, TCP congestion window of the sender as soon as a section of the RTO to significantly reduce the number of back and retransmits and lost packets. Channel errors can cause intermittent congestion window size by the sender to remain low, leading to low TCP throughput.

2. Mobility

Due to user mobility, Wireless networks are characterized by handoffs. Typically, handoffs may cause temporary disconnections, due to packet loss and delay. TCP congestion will suffer many losses and unnecessary congestion control mechanisms to deal with such calls. Handoffs are expected to occur more frequently in the next generation mobile network should be allowed to adapt to the increasing number of users of micro-cellular structure. TCP could be worse things that cannot gracefully handle handoffs. Similar problems may occur in wireless LAN, mobile users will encounter during user mobility which cause disruption.

2.3 Congestion Control Mechanism

2.3.1 TCP Tahoe

TCP Tahoe [14, 15] was the foremost algorithm to utilize three Congestion control Algorithms: slow start, congestion avoidance, and fast retransmit

2.3.2 TCP Reno

TCP Reno is the for the most part broadly embraced Internet TCP convention. It makes utilization of four Congestion control Algorithms: moderate begin, clog shirking, quick retransmit, and quick recuperation [16].

When bundle misfortune happens in a congested connection because of cradle flood in the middle person switches, either the sender gets three generation affirmations or the sender's retransmission timeout (RTO clock terminates). In the event of three generation ACKs, the sender enacts TCP quick retransmit and recuperation calculations and decreases its blockage window size to half. It then straightly builds clog window, practically identical to the instance of blockage shirking. This improve in transmission rate is slower than on account of moderate begin and calms blockage. TCP Reno quick recuperation calculation enhances TCP execution if there should be an occurrence of a solitary bundle misfortune inside of a window of information. Then again, execution of TCP Reno endures if there should arise an occurrence of complex parcel misfortunes inside of a window of information.

2.3.3 TCP NewReno

TCP NewReno [17] is an amendment of TCP Reno. It progresses retransmission process during the fast recovery phase of TCP Reno. TCP NewReno can perceive multiple packet losses. It does not outlet the fast recovery phase until all unacknowledged segments at the season of quick recuperation are recognized. Thus, as in TCP Reno, it surmounts diminishing the blockage window measure different times if there should be an occurrence of various parcel misfortunes. The staying three stages (moderate begin, blockage evasion, and quick retransmit) are like TCP Reno. TCP NewReno leaves quick recuperation subsequent to getting affirmation of every single unacknowledged fragment. It a while later sets clog window size to moderate begin limit and keeps up the blockage evasion stage. It retransmits the following section when it acknowledges a halfway affirmation. (Fractional affirmations are the affirmations that don't recognize every single exceptional parcel at the onset of the quick recuperation.) A trouble happens with New Reno when there are no bundle misfortunes however rather, bundles are reordered by more than 3 parcel arrangement numbers.

3.2.4 TCP SACK

SACK calculation [18, 19] permits a TCP collector to perceive out-of request fragments specifically as opposed to in total by recognizing the last suitably all together got section. The recipient recognizes parcels got out of request and the sender then retransmits just the missing information fragments as opposed to sending every unacknowledged portion. TCP Reno with SACK carries on comparably to TCP Tahoe and TCP Reno, which are vigorous if there should be an occurrence of out of request parcel landings. Then again, TCP with SACK shows signs of improvement execution if there should arise an occurrence of different parcel misfortunes. Amid the quick recuperation stage, SACK safeguards a variable called channel that speaks to the evaluated number of exceptional bundles the sender just sends new or retransmitted information when the assessed number of parcel in a switch is littler than the clog window. The funnel variable is augmented by one when the sender either sends another section or retransmits an old one. It is decremented by one when the sender gets the copy ACK with a SACK option [20].

3. *RELATED WORK*

Lijun Chen et. al. [3] proposed temporary traffic control for wireless networks, a combination of routing and scheduling. They refer to different types of networks and the ability to allocate resources in the network as the ratio between the generating utility problem previously with limitations, dual algorithm further increased to manage the network with a different channel and controlling device multi-rate for the high congestion control and quality of service in MANET.

Bhatia et al. [4] Proposed an agent based congestion control technique where the information about the network congestion is collected and distributed by the mobile agents. This algorithm was proposed to avoid congestion in the network. The routing protocol used is AODV routing protocol in which the mobile agents moves through the network and updates the routing table according to the node's congestion status. By the simulation results it is observed that this technique attains high throughput and packet delivery ratio with reduced delay and routing load as compared to other existing techniques.

Boraiah et al. [5] proposed congestion Adaptive AODV routing protocol (CA-AODV) is whenever multimedia based traffic such as voice, audio, video or text is transmitted over the network. CA-AODV is used to address the congestion issues considering routing overhead, delay and packet loss. This protocol ensures the availability of alternative routes along with the primary routes to reduce the routing overhead. If at any point of time, congestion occurs in the primary route between source node and destination node, the concerned node warns its previous node about congestion and an alternate route is selected for transmission to the destination node. This algorithm is concerned for real time communications and is useful for better performance in heavy traffic as well.

Islam et al. [6] proposed a multilevel congestion avoidance and control mechanism (MCCM) that develops both congestion avoidance and control mechanism to control the congestion predicament in an successful and efficient way. MCCM is proficient of finding an energy proficient path during route discovery process, make available longer lifetime of any developed route. The proficient admission control and selective data packet delivery mechanism of MCCM jointly triumph over the congestion problem at any node and thus, MCCM advances the network performance in term of packet delivery ratio, lower data delivery delay and high throughput. The consequences of performance evaluation section showed that, MCCM outperforms the existing routing protocols carried out in Network Simulator-2(NS-2).

Chen et al. [7] displayed shape for the helpful outline of clog control and media access control (MAC) for specially appointed remote systems. Utilizing conflict diagram and dispute framework, they planned asset circulation in the system as an adequacy expansion pickle with imperatives that emerge from dispute for channel access. They displayed two calculations that are circulated spatially, as well as all the more fascinatingly, they deteriorate oppositely into two convention layers where TCP and MAC mutually translate the framework issue. The essential is a primal calculation where the MAC layer at the connections produces blockage (conflict) costs in view of nearby combined source rates, and TCP sources alter their rates in view of the total costs in their ways. The resulting is a double sub-angle calculation where the MAC sub-calculation is actualized through booking connection layer streams as per the blockage costs of the connections. Worldwide merging properties of these calculations are demonstrated. This is an initial step towards a methodical way to deal with mutually plan TCP blockage control calculations and MAC calculations to create execution, as well as all the more noticeably, to make their collaboration more translucent.

4. *PROPOSED WORK*

4.1 Proposed Methodology

In the proposed methodology ACK scheme is modified for authentication of node in AODV protocol. By this type of approach it facilitate some misbehave activities of suspicious node in entire network. Before it various author proposed ACK-Based scheme but there was a common problem like overhead of huge amount of packets and node ambiguity. But in the proposed scheme secure channel control a generation of additional packets and also improved the AODV protocol performance. In the next section ACK-Based scheme, and its combine modified secure channel approach is discussed, which gives the high performance to the AODV protocol.

4.2 ACK-Based Approach

In here, when power transmission is used then ACK based scheme overcome the restriction of passive-feedback technique, for this an authentication mechanism is used to prevent the next hop from sending a forged ACK packet on behalf of the intended two hop neighbor. The chief shortcoming of this format is the enormous overhead. In order to decrease the overhead, the authors have proposed in [21] that each node asks its two hop neighbor to send back an ACK randomly rather than incessantly. Similarly, this extension also is unsuccessful when the two hop neighbor refuses to send back an ACK. In such circumstances, the requester node is unable to differentiate who is the malevolent node, its next hop or the requested node. The proposed technique has a reasonable overhead induced by the ACK sent back by the destination during preferred intervals of data transfer period. Throughout the data packets transmission, a flow of individual packets is transmitted at indiscriminate interims alongside the information. The reaction of these exceptional parcels summons the destination to convey an ACK from side to side various ways. The ACK bundles take numerous courses to gather the likelihood that all ACKs being dropped by the noxious hubs, furthermore to record for plausible misfortune because of softened courses or blockage up persuaded hubs. On the off chance that the source hub does not get any ACK parcel, then it gets to be aware of the vicinity of aggressors in the sending way. As a response, it shows a rundown of suspected malignant hubs to isolate them from the system. Every one of the hubs running an illustration in light of affirmation need to keep up a timeout (To) esteem. This timeout compares to an upper bound of the time that the sender hub needs to sit tight for the ACK to show up. The determination of this timeout worth is basic since a little esteem affects an expansive number of fake allegations and an extensive quality upgrades the memory required to store the active bundles for further correlations. Figure depicts a sample of the lower bound of the timeout worth kept up by hub A for the gathering of two jumps ACK from hub C. The timeout quality ought to be more noteworthy than the assessed edge (Th) esteem which can be calculated ascertained as as follows :

$$Th= T1\text{-}T2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

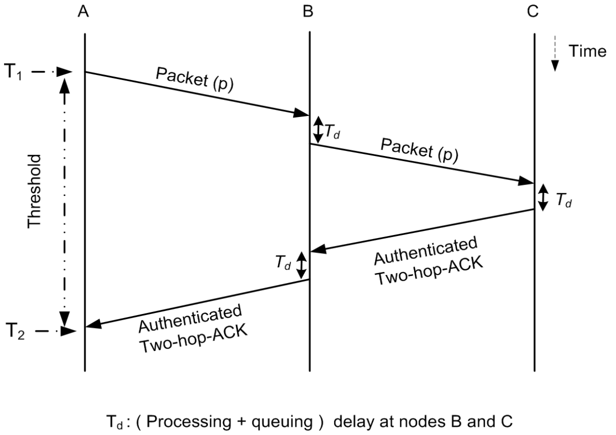

A
B
C
Time
T1
T2
Threshold
Packet (p)
Packet (p)
Td
Td
Td
Authenticated Two-hop-ACK
Authenticated Two-hop-ACK
Td : (Processing + queuing) delay at nodes B and C

Figure 4.1 Ack-Based Approaches

Where T1 and T2 are the sending (reception) time of the packet (ACK) respectively. This threshold is estimated for a successful transmission at MAC layer without any retransmission, which is not a realistic assumption in MANETs, thus the timeout value should satisfy the following condition

To > Th + (AV G RT × 1 hop delay)……………..(2)

Where AV G RT is the normal number of retransmissions of a bundle at MAC layer and 1 jump deferral is the one bounce transmission delay which incorporates parcel transmission delay, irregular Bakeoff postponement at the MAC layer and the preparing deferral.

4.2.1 Modified ACK-Based Scheme

In the modified scheme we also reduced the unnecessary 2 ACK, due to this there was a huge packets generated and they create ambiguity for the requested node and affect to the QOS, but in the proposed modified secure routing algorithm they maintain a state of node as well as a path state due to given request and response for maintaining a request packet ACK.

4.2.2 Secure Channel

Communication sessions in which a pair of parties begins by successively legitimated key-exchange protocol to acquire a shared session key, and afterward secure successive data transmissions among them through an authenticated encryption method based on the session key. We demonstrate that such a communication session congregates the notion of a secure channel protocol proposed by Canetti and Krawczyk [22] if and only if the underlying authenticated encryption mechanism meets two novel, easy definitions of security that we initiate, and the key-exchange protocol is secure. In other words, we diminish the secure channel requirements of Canetti and Krawczyk to effortless to utilize, stand-alone security requirements on the underlying authenticated encryption scheme. Additionally, we communicate the two new notions to existing security notions for authenticated encryption methods. We consider communication sessions in which a pair of parties begin by running an authenticated [22] key-exchange (KE) protocol to obtain a shared session key, and then secure successive data transmissions between them via an authenticated encryption scheme, a shared-key-based encryption scheme whose objective is to present both privacy and legitimacy, based on the session key.

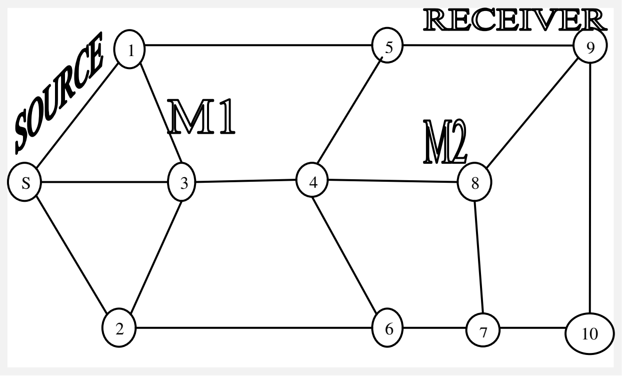


Figure 4.2 Secure Channel Display from Source to Receiver

4.3 Secure Channel Algorithm

Step 1: Set Sender Node = S
Step 2: Set Route Request = RREQ
Step 3: Set Random Sequence Number
Step 4: Set shared Key = SHKST

```
Step 5: Discovery route, with RRQ packet.
Step 6: Identifier for, random query identifier (RND) or sequence number (SQNO).
Step 7: Established Shared Key between Sender to Receiver.
            S $\longleftrightarrow$ R (SHKST);
Step 8: Sender (S) Sends a message authentication code (MAC) to Receiver (R).
          Active: (RND, SQNO + SHKST );
          MAC = HASH (S, R, RND, SQNO, SHKST);
Step 9: Accumulate IP address of intermediate nodes.
Step 11: Traversed packet from intermediate node .
          Tsavd packet == S or R;
Step 12: Intermediate nodes switch RREQ
Step 13: Intermediate node: Store RNDSQNO
          Inode: RNDSQNO + SHKST;
Step 14: Reject old RREQ;
          Flash: old RREQ;
          Jump to step 5;
Step 15: IF (Pkt > 1) THEN
                    //may received at receiver node via different path.
          ELSE
                    Receiver (R) generate MAC and verified by Sender (S).
Step 16: Route reply: (RST: n1, n2,…………nm);
Step 17: Receiver (R): Count MAC.
Step 18: R: Revert ACK to S via same/alternate route.
Step 19: Packet transmission continue.
```

5. EXPERIMENTAL RESULTS

5.1 Simulation Setup

The simulation of our proposed methodology is done in network simulator NS-2.34 [23] which is an open-source object-oriented discrete-event simulator for network research. The simulator is written in C++, with an OTcl (Object Tool Command Language) interpreter used as the command interface. We modeled network traffic using Constant Bit Rate (CBR) sources. A CBR traffic source provides a constant stream of packets throughout the whole simulation, thus further stressing the routing task. In each experiment, half the nodes in the network are CBR sources, and each source transmits 64-byte packets at a rate of 4 per second. We experimented with higher sending rates, packet sizes and number of sources. We omit those results, as they show similar trends, with the predictably higher effect of network congestion. In this thesis first we implemented AODV routing protocol for MANET under varying CBR (Various Parameters) and Scenes (Various Parameters) by varying the different parameters. After this we implemented malicious nodes in the network and then evaluated AODV routing protocol under the same conditions on which we evaluated AODV routing protocol for MANET. All the parameters have varied on the averages of five runs over different randomly generated mobility patterns.

Table 5.1 Simulation Setup

Simulation used	NS-2.34
Topology area	1200 X 1200
No. of Mobile Nodes	25
Max. No. of Connection	30
Simulation Time	200
Speed	10-20 m/sec
Communication Link Capacity	10 Mbps
Traffic Intensity	45,85,95,180
Routing Protocol	AODV

5.2 Result Analysis

The analysis of the proposed work is performing by using the performance measuring parameter such as Routing load , Packet delivery ratio and Throughput. The experimental results of proposed work for packet delivery ratio with existing method is depicted through table and shown in graph.

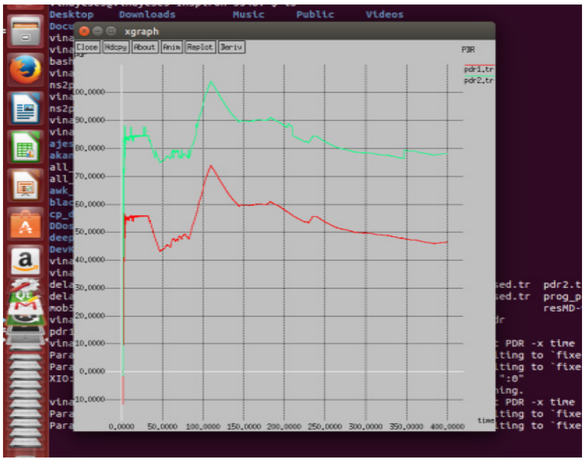
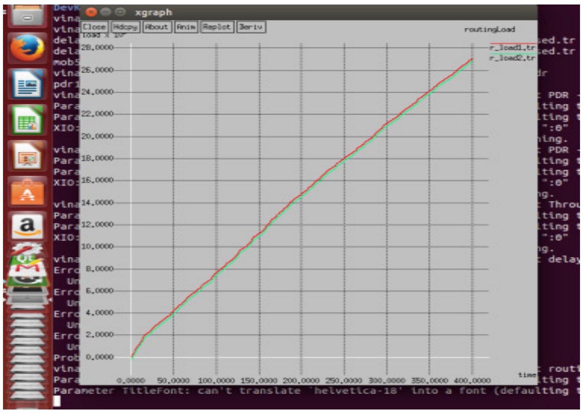
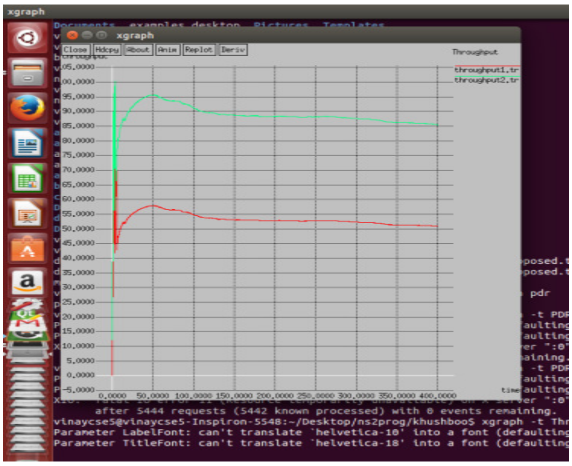


Fig. 5.1 Graph of PDR between existing and proposed method



```
Parameter TitleFont: can't translate `helvetica-18' into a font (defaulting
```

Fig. 5.2 Graph of routing load between existing and proposed method



```
     after 5444 requests (5442 known processed) with 0 events remaining.
vinaycse5@vinaycse5-Inspiron-5548:~/Desktop/ns2prog/khushboo$ xgraph -t Thr
Parameter LabelFont: can't translate `helvetica-10' into a font (defaulting
Parameter TitleFont: can't translate `helvetica-18' into a font (defaulting
```

Fig. 5.3 Graph of throughput between existing and proposed method

After simulation the proposed work among different parameter Routing load, Throughput and PDR it is found that the outcomes of our method is much better.

6. CONCLUSION

Due to the mobility behavior of ad hoc network packet loss and congestion occurs. In this paper we proposed modified ACK based scheme to control the congestion over the network and also develop a secure channel algorithm to follow the step to deliver the packet. The experimental result of our proposed scheme outperform much better than the existing approach. The comparison of the proposed is done with respect to PDR, Throughput and routing load in which the packet delivery rate of our method is approximate 35% more and throughput 50% higher than the existing method.

REFERENCES

[1]. Shraddha Raut, Sd Chede: “Detection and Removal of Black hole in Mobile Ad-hoc Network (MANET)”, International Journal of Electrical and Electronics Engineering (IJEEE), ISSN (PRINT): 2231 – 5284 Vol-1 Iss-4, 2012.

[2]. A K Mourya, N Singhal, „Managing Congestion Control In MobileAd-Hoc Network Using Mobile Agents‟, International Journal ofComputerEngineering& Applications, Vol. IV, Issue I/III, 2013.

[3]. Chen, L., Lowy, S.H., Chiangz, M., Doyley, J.C.: Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks. Proc., IEEE, 25th International Conference on Computer Communication, INFOCOM. pp 1 - 13.(2007).

[4]. Bandana Bhatia, Neha Sood, “AODV Based Congestion Control Protocols”. International Journal of Computer Science and Information Technology 2014.

[5]. Boraiah Ramesh, “CA-AODV Congestion Adaptive AODV Routing Protocol for Streaming Video in Mobile Ad-hoc Networks”. I.J Communications, Network and System Sciences 2008.

[6]. Manowarul Islam, Abdur Razzaqu, Ashraf Uddin and A.K.M Kamrul Islam “MCCM: Multilevel Congestion Avoidance and Control Mechanism for Mobile Ad Hoc Networks” I.J. Information Technology and Computer Science, 2014, 06, 9-18 Published Online May 2014 in MECS.

[7]. Lijun Chen, Steven H. Low and John C. Doyle “Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks”.

[8]. Abinasha Mohan Borah, Bobby Sharma and Manab Mohan Borah “A Congestion Control Algorithm for Mobility Model in Mobile Ad-hoc Networks”, International Journal of Computer Applications (0975 – 8887) Volume 118 – No.23, May 2015.

[9]. S. A. Jain and Sujata K. Tapkir, "A Review of Improvement in TCP congestion Control Using Route Failure Detection in MANET”, Network and Complex Systems, Vol 2, No.2, pp.9-13, 2012.

[10]. M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018 (Proposed Standard), Oct. 1996.

[11]. Yung Yi and Sanjay Shakkottai. Hop-by-hop Congestion Control over a Wireless Multi-hop Network, 0-7803-8356- 7/04/$20.00 (C) 2004 IEEE.

[12]. Yi-Cheng Chan and Hon-JieLee,"A Hybrid Congestion Control for TCP over High Speed Networks. 2012 Sixth International Conference on Genetic and Evolutionary Computing. 2012.

[13]. Lien, Y.N., Hsiao, H.C.: A New TCP Congestion Control Mechanism over Wireless Ad Hoc Networks by Router-Assisted Approach. 27th IEEE International Conference on Distributed Computing Systems Workshops. (2007).

[14]. V. Jacobson and M. J. Karels, "Congestion avoidance and control", In ACM Computer Communication Review; Proceedings of the Sigcomm’88 Symposium, volume 18, pages 314–329, Stanford, CA, USA, August 1988.

[15]. Hanaa A. Torkey, Gamal M. Attiya and I. Z. Morsi, "Performance Evaluation of End-to-End Congestion Control Protocols", Menoufia journal of Electronic Engineering Research (MJEER), Vol. 18, no. 2, pp. 99-118, July 2008.

[16]. M. Allman, V. Paxson, and W. Stevens. RFC 2581 - TCP Congestion Control. The Internet Society, 1999.

[17]. Hanaa A. Torkey, Gamal M. Attiya and I. Z. Morsi, "Enhanced Fast Recovery Mechanism for improving TCP NewReno", Proceedings of the 18th International Conference on Computer Theory and Applications (ICCTA08), pp. 52-58, Alexandria, Egypt, 11-13 October 2008.

[18]. V. Jacobson and R. Braden. RFC 1072 – TCP Extensions for Long Delay Paths. October 1988.

[19]. Beomjoon Kim, Dongmin Kim, and Jaiyong Lee, "Lost Retransmission Detection for TCP SACK", IEEE COMMUNICATIONS LETTERS, VOL. 8, NO. 9, September 2004.

[20]. V. Jacobson, "Modified TCP congestion avoidance algorithm", url:ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt.

[21]. Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges Soufiene Djahel, Farid Na¨ıt-abdesselam, and Zonghua Zhang, IEEE 2010.

[22]. Secure Channels based on Authenticated Encryption Schemes: A Simple Characterization Chanathip Namprempre August 29, 2002.

[23]. N.Drakos and R.Moore, ns2 -The Manual (formerly Notes and Documentation), 1999. [Online]. Available: http://www.isi.edu/nsnam/ns/doc/